Experience with a Class Library for Organizational Modeling and Problem Solving

B. R. GAINES

Knowledge Science Institute, University of Calgary, Calgary, Alberta, Canada T2N 1N4

ABSTRACT

The implementation architecture of a knowledge representation server as a class library in an object-oriented language is described. The graphic editing tools developed to support ease of interactive editing of the objects in the knowledge representation system are illustrated. It is shown that object-oriented paradigms are effective in knowledge representation for organizational modeling and problem solving. They allow system architectures to be developed that combine centralized repositories and distributed heterogeneous subsystems to support new forms of organizational integration. © 1993 John Wiley & Sons, Inc.

INTRODUCTION

Organizations are complex structures of human roles and relations, information, financial and material flows, power and accountability relations, skills, and knowledge (Scott, 1981). However, these structures have been only implicitly represented in current information technology. This is not by choice but because our "high-level" programming languages, while a major advance on machine code, have focused on efficient data processing rather than organizational modeling (Scheer, 1989). Normalized relational, and entity–relation, databases have some degree of corporate modeling capability, but their limited passive representations may be seen as capturing data about the activities of an organization rather than modeling its procedures, knowledge, and skills (Gaines, 1988). It is only in recent years that the combination of advances in computing power and software engineering techniques has allowed practical application of object-oriented paradigms in which modeling and data processing can be effectively combined (Gupta, 1991).

There are now a number of major research projects concerned with the development and application of object-oriented modeling methodologies and tools to organizational information systems (Afsarmanesh and McLeod, 1989; Sciore, 1986) and to significant subsystems such as documents (Güting, Zicari, and Choy, 1989) and electronic mail (Lee and Malone, 1990). In particular, the ESPRIT TO-DOS project for the design of office information systems has adopted the conceptual modeling and object-oriented design approaches on a large scale (Pernici et al., 1989; Pernici and Rolland, 1990). The routine application of the object-oriented modeling approach is yet to come, although the mainstream expert system shells such as ART, KEE, and

Received February 22, 1993; accepted May 17, 1993
Integrated Computer-Aided Engineering, 1(2) 93–107 (1993)
© 1993 John Wiley & Sons, Inc. CCC 1069-2509/93/020093-15

NEXPERT all provide object-oriented knowledge representation environments and there is convergence between artificial intelligence, hypermedia, and knowledge base technologies (Gupta, 1991; Parsaye et al., 1989).

Major questions remain as to whether the next generation of large-scale applications should be based on corporate knowledge repositories as extensions of current databases (Davis, 1989) or on distributed, heterogeneously integrated knowledge bases as extensions of networked personal computing environments. Probably a combination of both is required because computational complexity limitations (Nebel, 1990) in deductive databases severely limit the scalability of current expert systems to become "corporate minds," yet the anarchy of uncontrolled distribution would lead to precisely those problems of inconsistency and unreliability of data that the repository approach seeks to avoid (Sagawa, 1990).

This article reports some studies of organizational modeling and problem solving based on the hypothesis that it is the user interface to corporate knowledge bases that will be the primary bottleneck in their effective implementation. Knowledge, as philosophers have noted (Runes, 1972), is somewhere between facts and opinions. In computational terms, effective knowledge-based systems require a dynamic interplay between the database of facts and the knowledge processes of skilled human experts. Providing a high-quality interface for that interplay is crucial to the continuous management of the uncertainties, errors, conflicts, and continually changing situations that make up the knowledge environment of an organization.

The approach taken in the studies reported here has been to implement recent advances in the design of well-specified knowledge representation and reasoning systems as a lightweight knowledge representation server and use the object-oriented programming paradigm to surround the server with small functional modules supporting a variety of applications (Gaines, 1991a, 1991b). The server is sufficiently lightweight that multiple instantiations of it can exist on personal workstations yet sufficiently fast and powerful to cope with knowledge structures involving thousands of concepts and tens of thousands of entities.

It is assumed that the attributes and relations of entities are in large part extracted from a corporate data repository, that high-level concept definitions modeling the organization and its procedures are similarly extracted from a corporate knowledge repository, that many problem-solving concept/rule structures are available in corporate libraries, and that many others will be generated as required within local environments. It is assumed that one major function of human use of this data and knowledge is to validate it and keep it updated, effective, and appropriate to current situations. Hence the presentation of the data and knowledge structures in an easily understandable and editable form is a major design priority.

The approach taken to data and knowledge presentation, retrieval, inference, and editing has been to develop a formal visual language supported through graphic editors providing a visualization and drawing environment similar to that of drawing tools and with strong attention to principles of cognitive ergonomics (Gaines, 1991c). More conventional user interfaces to spreadsheets, databases, word processing, and accounting packages are also provided largely for factual material, but the primary interface to the knowledge structures representing conceptual material is through the visual language. The language supports visual libraries of conceptual and problem-solving structures that can be linked with specifically developed knowledge structures so that the balance between centrally and locally managed knowledge structures can be varied and managed as required.

KSSn ARCHITECTURE

Knowledge Support System n (KSSn) is the latest in a series of developments deriving from our initial implementation of KSS0 (Gaines, 1988b) and KSS1 (Shaw and Gaines, 1987). These early knowledge support systems focused on knowledge acquisition and conceptual modeling and have been extended through heterogeneous integration to offer close integration with hypermedia and expert system shells (Gaines, Rappaport, and Shaw, 1989; Linster, 1991b). One problem with research and applications in this area is that of rapid development in techniques, tools, and technologies. Systems become obsolescent in a short time and a monolithic architecture is difficult to maintain and enhance. To cope with this, we have completely reengineered KSSn as a C++ class library implementing KRS, a KL-ONE-like knowledge representation server, and a set of associated functional modules for knowledge elicitation, text analysis, empirical induction, graphic knowledge base editing, and so on.

This approach is based on two recent developments that open up new possibilities for meaningful empirical research in artificial intelligence: new in-

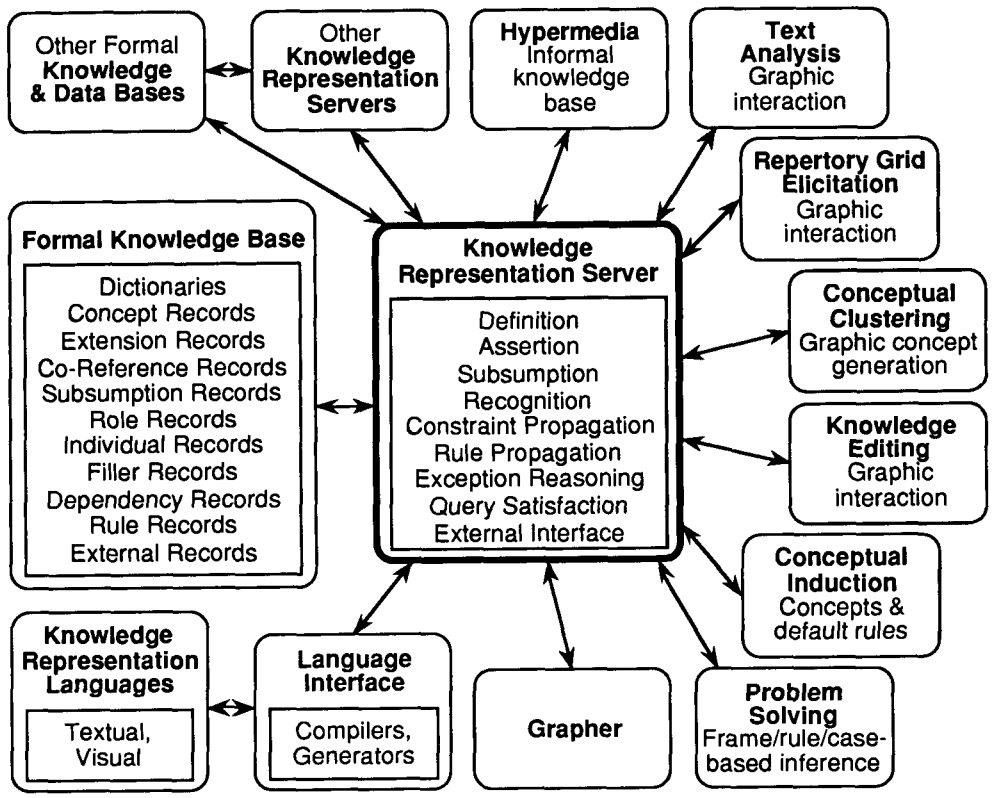

Figure 1 Architecture of the knowledge representation system.

sights into the complexity issues affecting knowledge representation and reasoning (Brachman and Levesque, 1984; Schmidt-Shauβ, 1989) and the development of object-oriented programming environments offering precise definition and efficient implementation of extensible abstract data types (Shriver and Wegner, 1987). The advances in artificial intelligence allow requirements specifications to be developed for knowledge representation and reasoning systems with known, and practically applicable, performance curves (Borgida et al., 1989). The advances in software engineering allow these specifications to be implemented cleanly in such a way that their conformance with the requirements can be verified and their algorithms achieve the best possible performance.

In practical terms, the series of knowledge representation schemes and tools commencing with KL-ONE (Brachman and Schmolze, 1985), developing through KRYPTON (Brachman, Gilbert, and Levesque, 1985), and currently culminating in systems such as CLASSIC (Borgida et al., 1989) and LOOM (MacGregor, 1988) has reached a maturity of technology that offers the promise of knowledge representation "utilities" or "services" in Levesque's (1984) terminology. The logical foundations of the subsumption relation, techniques for its correct and complete calculation, and the interaction between representation power and the tractability of subsumption algorithms have been widely studied (Brachman and Levesque, 1984; Nebel, 1990; Schmidt-Shauβ, 1989). It is now feasible to develop knowledge representation servers on a par with floating-point arithmetic units and numeric libraries as software (and perhaps ultimately hardware) modules with well-defined functionality and fast, reliable performance. As with arithmetic units, such servers by no means solve all the problems of a particular application domain, but they do greatly reduce the burden of system development, allowing effort to be focused on the specifics of particular systems rather than on what should be general utilities.

Figure 1 shows the architecture of KSSn as a family of modules attached to the knowledge representation server, KRS. The modules are (reading clockwise from the top left):

- Interface modules to other knowledge bases and servers, including databases.
- A hypermedia module allowing informal knowledge structures in text and images to be

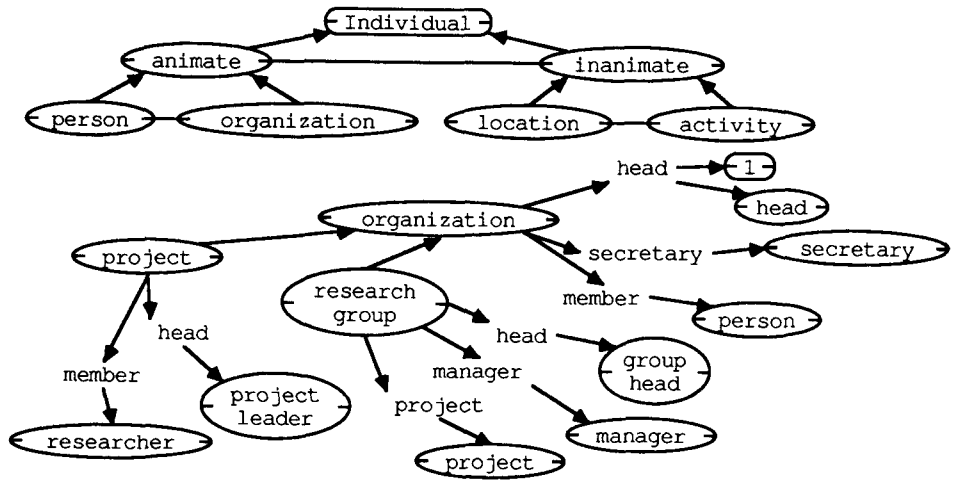

Figure 2 Top-level and organizational ontologies in KDraw visual language.

captured, accessed, and linked. The linkage structure itself is held as a knowledge base.

- A text analysis module allowing documents to be analyzed in terms of word usage and associations between significant words to be graphed—based on TEXAN in KSS0. This enables protocols and technical documents to be used to initiate knowledge acquisition.
- A repertory grid expertise transfer module allowing graphic definition of concepts and graphic creation and editing of individuals—based on the elicitation screens of KSS0.
- A conceptual clustering module allowing interactive definition of new concepts—based on the hierarchical and spatial clustering from KSS0.
- A knowledge editing module allowing the interactive development and editing of knowledge structures through a visual language.
- A conceptual induction module creating rules about specified subsets of individuals and transforming them to a minimal set of concepts and default rules—based on the induct algorithm.
- A problem-solving module supporting frame-, rule-, and case-based inference from the knowledge structures.
- A grapher laying out specified parts of the concept subsumption graph, concept structures, and individual structures—based on an incremental layout algorithm that can be used interactively to support the production of clear visual knowledge structures.
- A language interface accepting and generating definitions and assertions in formal knowledge representation languages, both textual and visual.

The knowledge representation services of KRS, the central server module, correspond to those of CLASSIC (Borgida et al., 1989), augmented with inverse roles, data types for integers, reals, strings,

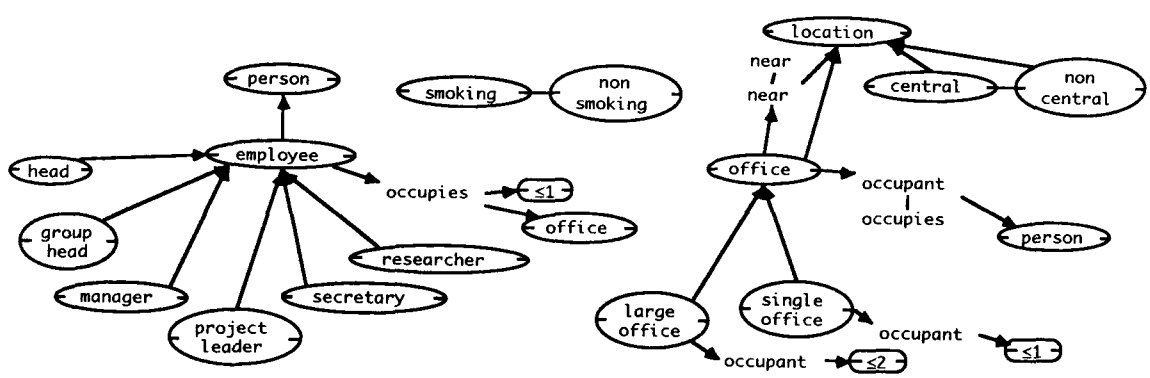

Figure 3 Employee and office ontologies.

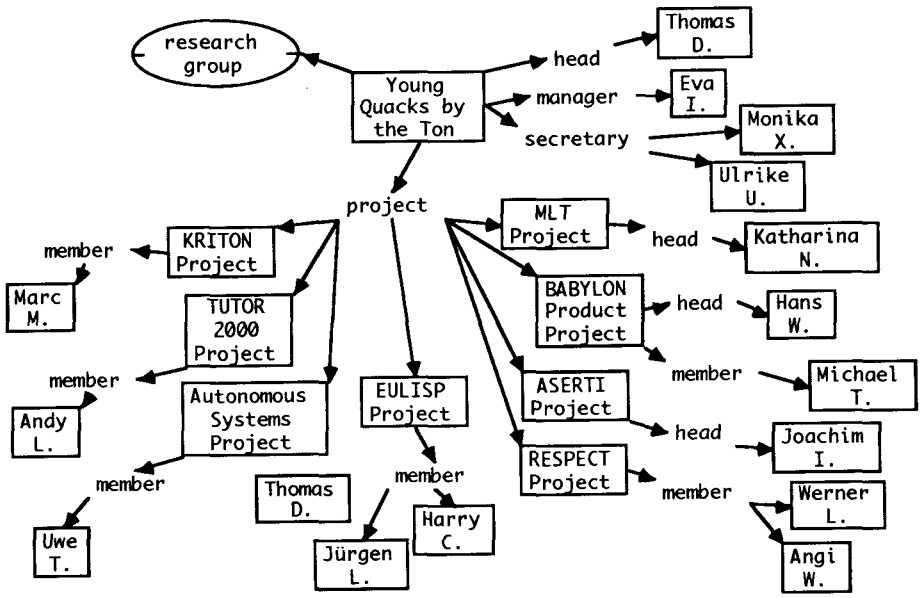

Figure 4 Facts asserted about projects and employees as an organization chart.

and dates, and with rule representation that allows one rule to be declared an exception to others. For the purposes of this article, KRS may be seen as providing a fast and principled implementation of a frame/rule knowledge representation and inference engine capable of operating with large knowledge bases.

KDRAW VISUAL LANGUAGE

An important component of KSSn in the context of this article is the knowledge editing through graphic interaction component, KDraw, at the right of Figure 1. This is a drawing tool designed for ease of use that provides a visual structure editor for semantic networks representing classes, objects, and rules in KRS. Nosek and Roth (1990) demonstrated empirically that the visual presentation of knowledge structures as semantic nets leads to more effective human understanding than does textual presentation of the same structures. We have developed a formal visual language that corresponds exactly to the underlying algebraic semantics of KRS that has remarkably few visual primitives and is easily learned and understood (Gaines and Shaw, 1990).

Visual representation of knowledge structures has been common since the early development of diagrams and taxonomies, and was associated with semantic networks in the early days of artificial intelligence (Quillian, 1968). The early development of such nets resulted in criticisms that the semantics of particular diagrams was not well defined (Brachman, 1977; Woods, 1975). Nodes, arcs, and their labels could be used freely and ambiguously and diagrams were subject to differing interpretations. In the 1970s, there were proposals for network formalisms with well-defined semantics (Brachman, 1979; Cercone and Schubert, 1975; Fahlman, 1979). However, these preceded two important developments in computing: first, the ubiquity of personal workstations with high-resolution graphics supporting visual languages as operational editors (Glinert, 1990); second, the studies of complexity issues in knowledge representation noted above, leading to the simplified and tractable semantics of CLASSIC.

In light of these developments, semantic networks were reexamined as formal visual languages from two perspectives:

- *Cognitive ergonomics.* Is it possible to create visual languages that are simple and natural to use in operational form as interactive structure editors?
- *Formal semantics.* Is it possible to intertranslate (i.e., translate unambiguously in both directions) between these languages and knowledge representation formalisms with well-defined semantics?

Note that the second question is deliberately phrased to avoid mixing the issues of knowledge representation, logic, intensional and extensional

semantics, and so on, which cast doubt upon the utility of early semantics networks, with the issues of visual representation. It leaves the deep semantic issues in the province of formal knowledge representation where they belong and assesses the semantic validity of a visual language by its intertranslatability with established formalisms (Mackinlay and Genesereth, 1984).

Computer production of visual forms of knowledge represented in a computer has been a topic of research since the early days of knowledge representation research (Schmolze, 1983) and a feature of many research systems (Kindermann and Quantz, 1989) and commercial products. Some expert system shells and knowledge acquisition tools have used the power of modern workstations to provide continuously updated graphs of some aspects of a knowledge structure being entered textually. Abrett and Burstein's (1988) KREME system graphically displays the computed subsumption relations between concepts so that those entering knowledge structures can see the consequences of definitions and detect errors due to incorrect or inadequate definitions. However, KREME does not support graphic knowledge entry or editing.

The KDraw design has drawn upon this previous research and experience to develop the visual syntax and underlying semantics of a visual language for term subsumption knowledge representation languages in the KL-ONE family. It focuses on the use of the language to enter and edit knowledge visually and on its application in a highly interactive graphic oxstructure editor. However, the language is also well suited to the display of knowledge structures, and the system includes a grapher using Watanabe's (1989) heuristics for the layout of complex graphs.

KDraw is described in detail elsewhere (Gaines, 1991c) and the best way to introduce it here is through examples drawn from the room allocation application described in the next section. Figure 2 shows the top-level conceptual structures of an organizational domain. The visual language used in Figure 2 is precisely defined. Concepts are ovals, primitive concepts ovals with small horizontal lines inside each side, individuals rectangles, roles unboxed text, rules rounded-corner boxes, and constraint expressions rounded-corner boxes with small horizontal lines. Lines without arrows connecting primitive concepts denote that the concepts are disjoint and connecting roles that they are inverse. The interpretation of the arrows in the editor is overloaded but well-defined by the types of the objects at their head and tail, for example,

concept → concept	definitional subsumption
concept → role → concept	definitional role with conceptual constraint
concept → role → constraint	definitional role with extensional, cardinality, or numeric constraint
constraint → individual	extensional constraint
individual → concept	asserted constraint on individual
individual → role → individual	asserted value of role for individual

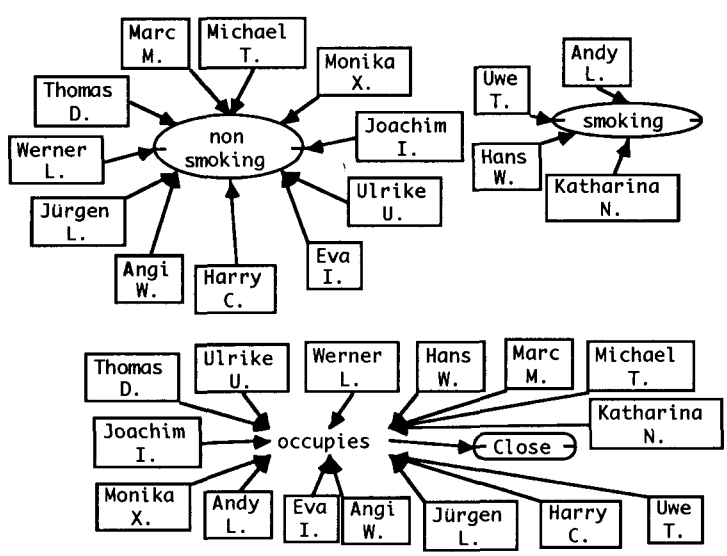

Figure 5 Facts asserted about employees.

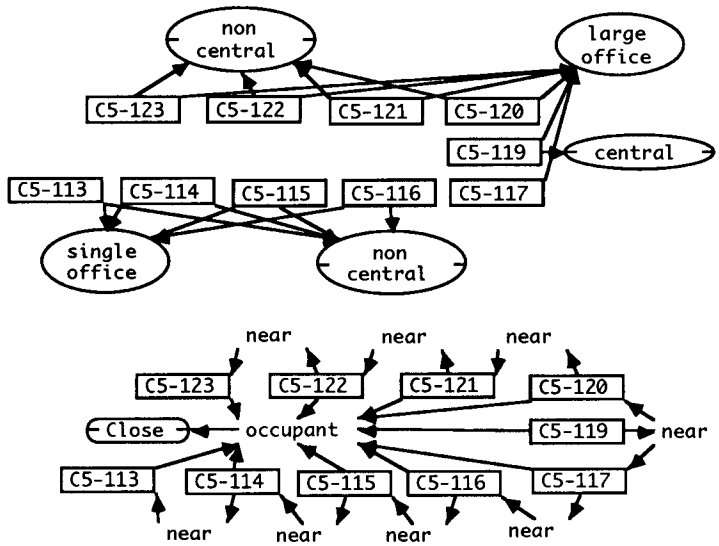

Figure 6 Facts asserted about rooms as a room layout plan.

concept → rule → concept — production rule
rule → rule — first rule exception to second

Thus, the top knowledge structure in Figure 2 defines "animate" and "inanimate" to be disjoint primitive concepts of type "Individual," "person" and "organization" to be disjoint concepts inheriting from "animate," and "location" and "activity" to be disjoint concepts inheriting from "inanimate." The lower knowledge structure further defines "organization" (note that nodes may be freely duplicated for the sake of visual appearance) to have the role "head" filled by exactly one individual of type "head," to have the role "secretary" possibly filled by individuals of type "secretary," and to have the role "member" possibly filled by individuals of type "person." A project, because it is shown to inherit from "organization," also has these roles and constraints but is shown to be further constrained to have its "head" role filled by a "group head" and its "member" role filled by individuals of type "researcher."

It is appropriate to note here that the underlying knowledge representation inference engine is deductive so that defining an individual as a "project" and then filling its "member" role with "Marc M." will automatically lead to "Marc M." being inferred to be a "researcher." Figure 3 shows how some of the concept definitions used in Figure 2 are expressed visually together with some other features of the language. For example, "occupant" and "occupies" are defined to be inverses, as are "near" and "near." Hence, if "Marc M." is defined to fill the "occupant" role of office "CS-120" then it will be inferred that "CS-120" fills the "occupies" role of "Marc M."

Figure 4 shows the way in which facts may be asserted about individuals in the visual language. For example, the "RESPECT Project" is shown to be one of the fillers of the "project" role of "Young Quacks by the Ton," which is defined to be an instance of a "research group." This concept instantiation has the consequence that all the conceptual constraints shown in Figures 2 and 3 will be inherited appropriately by individuals defined in Figure 4. For example, it will be inferred that "Thomas D." is "group head," "head," "person," and "animate" and is the only filler of the "head" role of "Young Quacks by the Ton." Figures 5 and 6 supply further facts about individuals and rooms that will be needed in the problem-solving activity.

What has been described so far may be seen as yet another semantic network presentation of knowledge structures as is common to many of the conceptual and object-oriented design articles cited earlier. However, there are a number of significant differences. First, the diagrams shown are fully operational. They were created in the KDraw graphic structure editor and compile directly into knowledge structures in KRS. Second, the visual language used is completely formally defined and intertranslatable with the underlying KL-ONE knowledge structures. Automatic graphing of the compiled structures in KRS results in these same diagrams except for layout information (the layout may make the knowledge

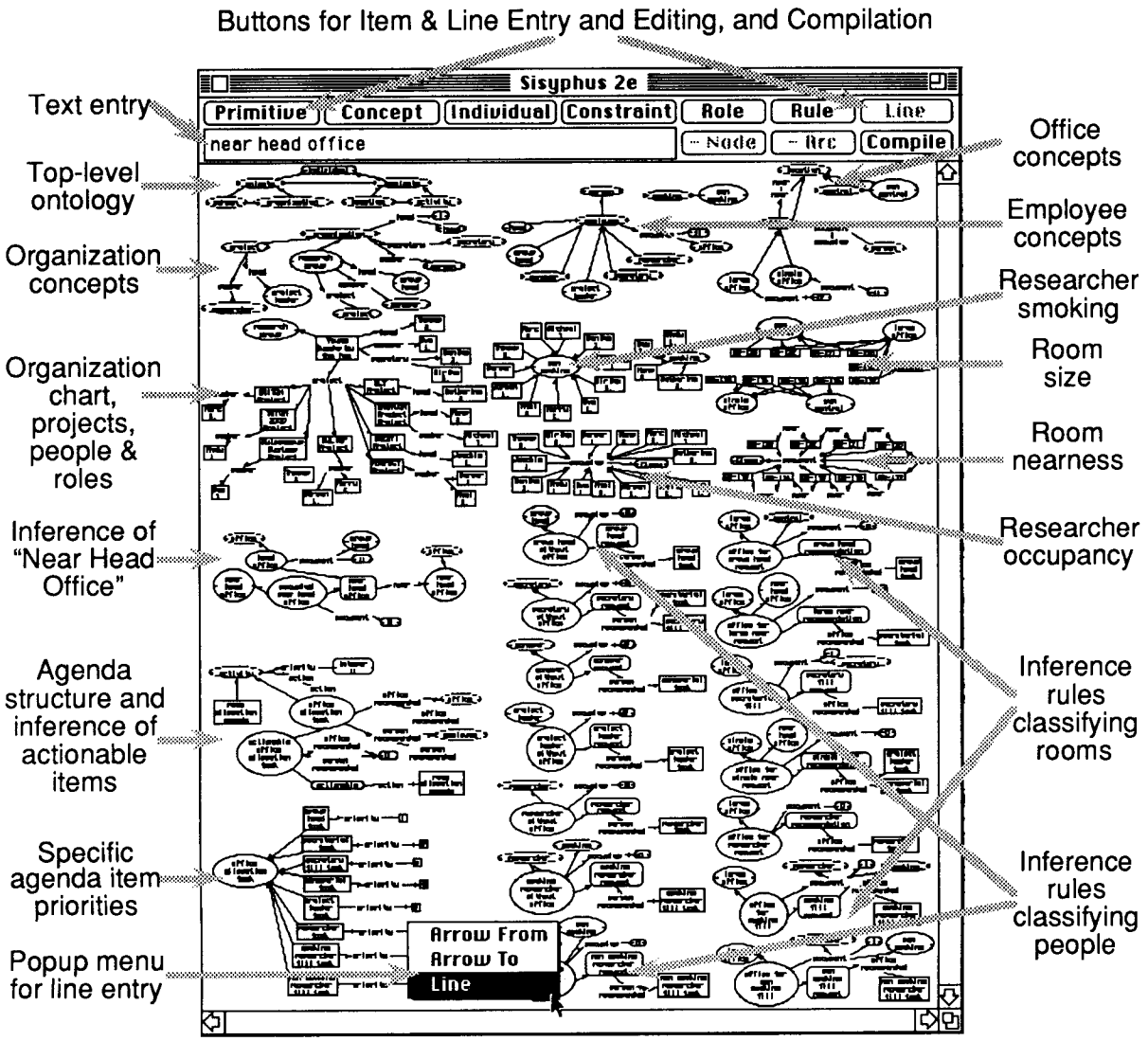

Figure 7 Problem, its solution, and example dataset expressed in KDraw visual language.

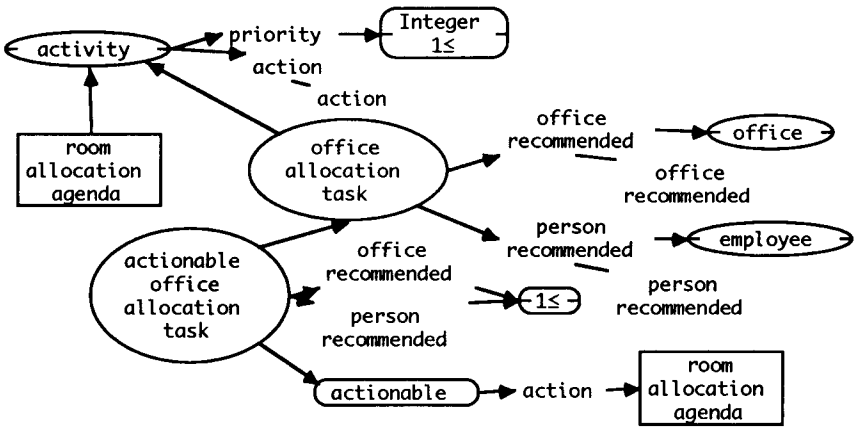

Figure 8 Concept definitions for agenda mechanism and rule putting items on it.

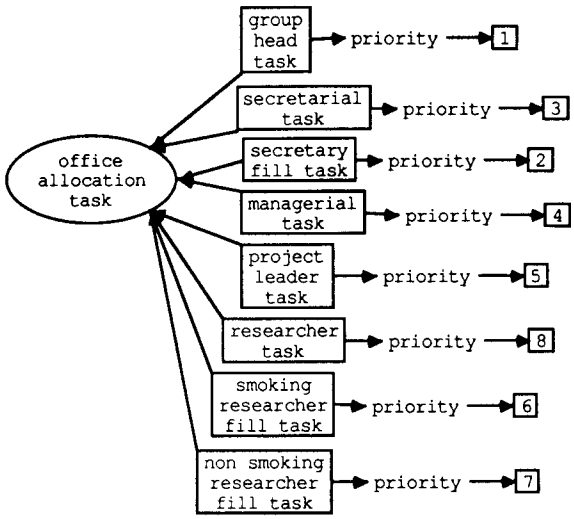

Figure 9 Priorities asserted for possible agenda items.

more readily assimilable by people but is not regarded as part of the formally represented knowledge). Third, the freedom in layout has been used to create knowledge structures that are natural to the people involved. Figure 4 looks like an organization chart. Figure 6 is based on the actual room layout in the building. Fourth, editing these structures changes the data and knowledge structures and hence any related problem-solving activity. For example, at the top of Figure 6 room "CS-119" is the only room asserted to be "central." If this appears to be restrictive when the room allocation rules are run, then it is easy for the user to add arrows from "CS-117" and "CS-120" to "central" and see what changes result in the room allocation behavior.

Figure 7 shows the complete knowledge, data, and problem-solving structures in the visual language as a screen dump from the KDraw editing window. A "fit-to-size" command was used to get the structures into a small diagram illustrating the main features of the interface. However, the structures shown in full in Figures 2–6 at the top of the screen are apparent.

Human–computer interaction in the editor is modeled on Apple's MacDraw with additional features appropriate to the language such as arcs remaining attached to nodes when they are dragged. A popup menu as shown at the bottom of Figure 7 allows connecting lines to be entered easily. The syntax of possible node interconnections and constraint expressions is enforced—it is not possible to enter a graph that is syntactically incorrect. Cut and paste of graphs and subgraphs is supported, and popup menus allow nodes to be connected with minimum effort. Updates are efficient and graphs with over 1000 nodes can be manipulated interactively. Scroll, zoom, and fit-to-size facilities allow large data structures to be navigated easily. However, partitioning data structures over several screens is encouraged and has proved practical in managing large knowledge structures.

SOLUTION OF A ROOM ALLOCATION PROBLEM

KDraw may be used just to support the entry of conceptual structures and facts in a knowledge base. However, it is capable of going beyond this and accepting problem-solving knowledge structures in the form of concepts defining the premise and conclusion of rules. The rule structure provided in KDraw and KRS is itself powerful in both its representational and inference capabilities in supporting exceptions and defaults (Gaines, 1991b). The following example extends that described above to illustrate these features.

The problem is one of room allocation derived from an ESPRIT project (Voβ et al., 1990) that has recently been placed in the public domain as part of Project Sisyphus. Sisyphus is a research

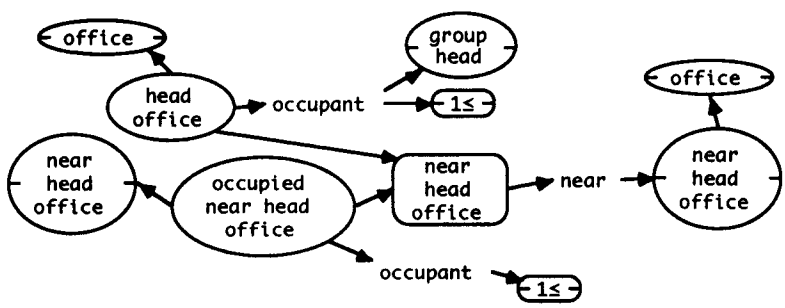

Figure 10 Rule inferring "near head office" and propagating it as offices fill.

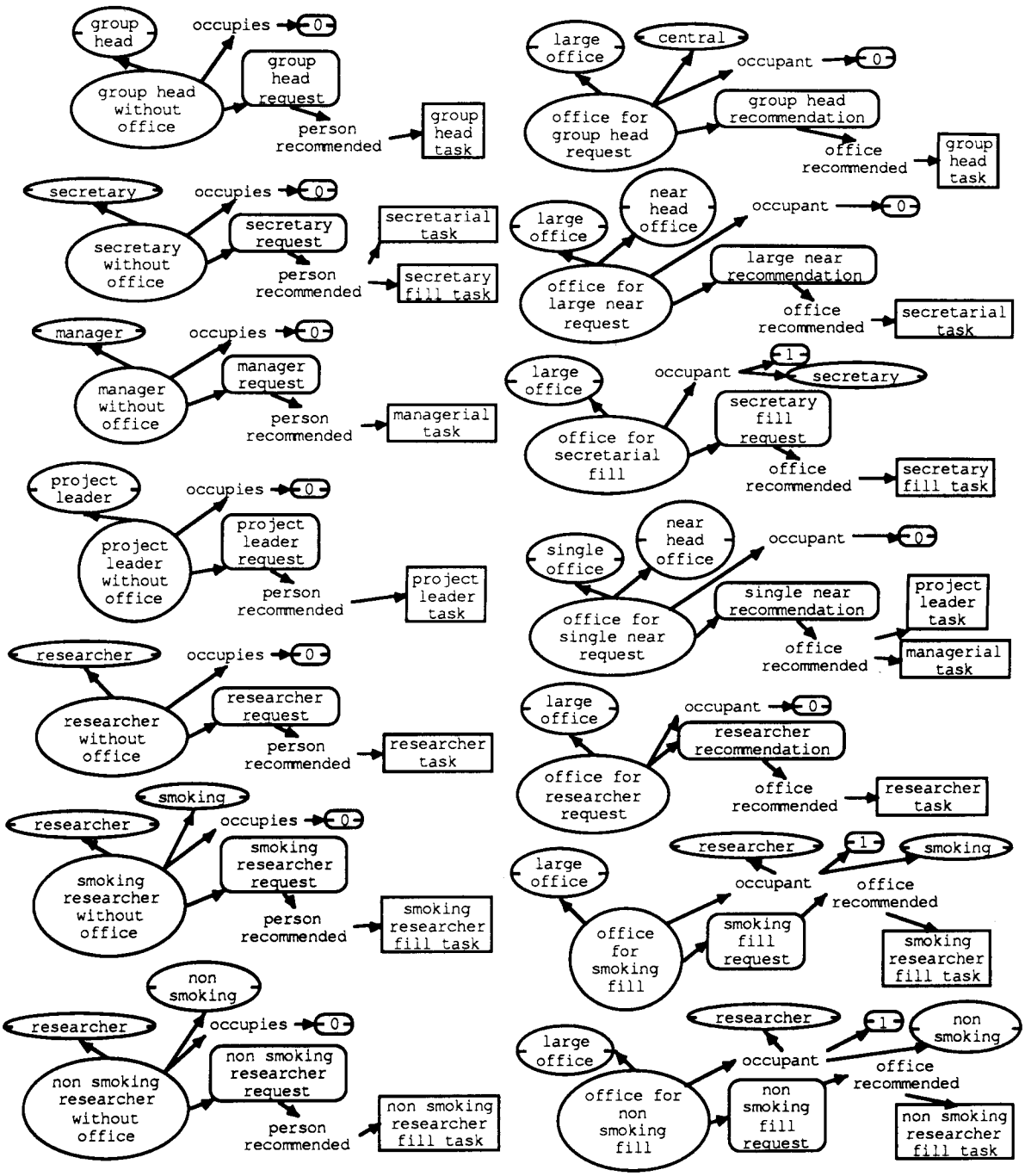

Figure 11 Rules classifying people (on left) and offices (on right).

program to encourage international collaboration in knowledge-based system development developed initially by the European Knowledge Acquisition Workshop in 1989. A number of problem datasets have been made available through Sisyphus, and a major part of the EKAW'91 program was devoted to reports on the solution of these problems using different approaches and techniques (Linster, 1991a). Some 16 research groups internationally participated and the ex-

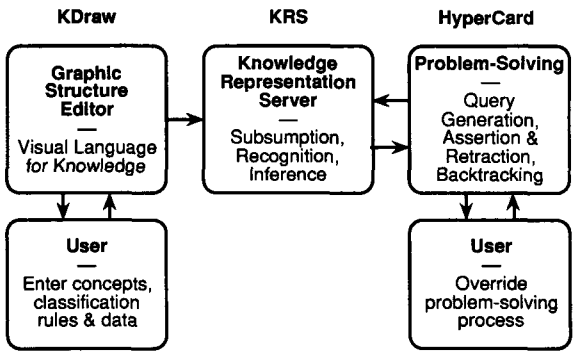

Figure 12 System architecture: KSSn subsystems, functions, and user interaction.

ample used here is one provided by the GMD in Bonn and coordinated by Marc Linster.

The problem is to take sample protocols from an expert making room assignments to researchers occupying a new building. The protocols are based on cases and involve the expert in explaining each decision as it is made. The detailed problem definition and knowledge acquisition procedures will not be described in detail in this article, but the results will be used to show the system described operating in a problem-solving mode.

The rules derived from the expert protocols are, in order of declining priority:

If an office is large and central and the Head of Group is without an office suggest the allocation.

A large office already occupied by a secretary should be allocated to another secretary.

A large office near the Head of Group is suitable for the secretaries.

A single office near the Head of Group is suitable for the Manager.

A single office near the Head of Group is suitable for the Project Leaders.

A large office with a smoking researcher occupant should be allocated to another smoking researcher.

A large office with a nonsmoking researcher occupant should be allocated to another nonsmoking researcher.

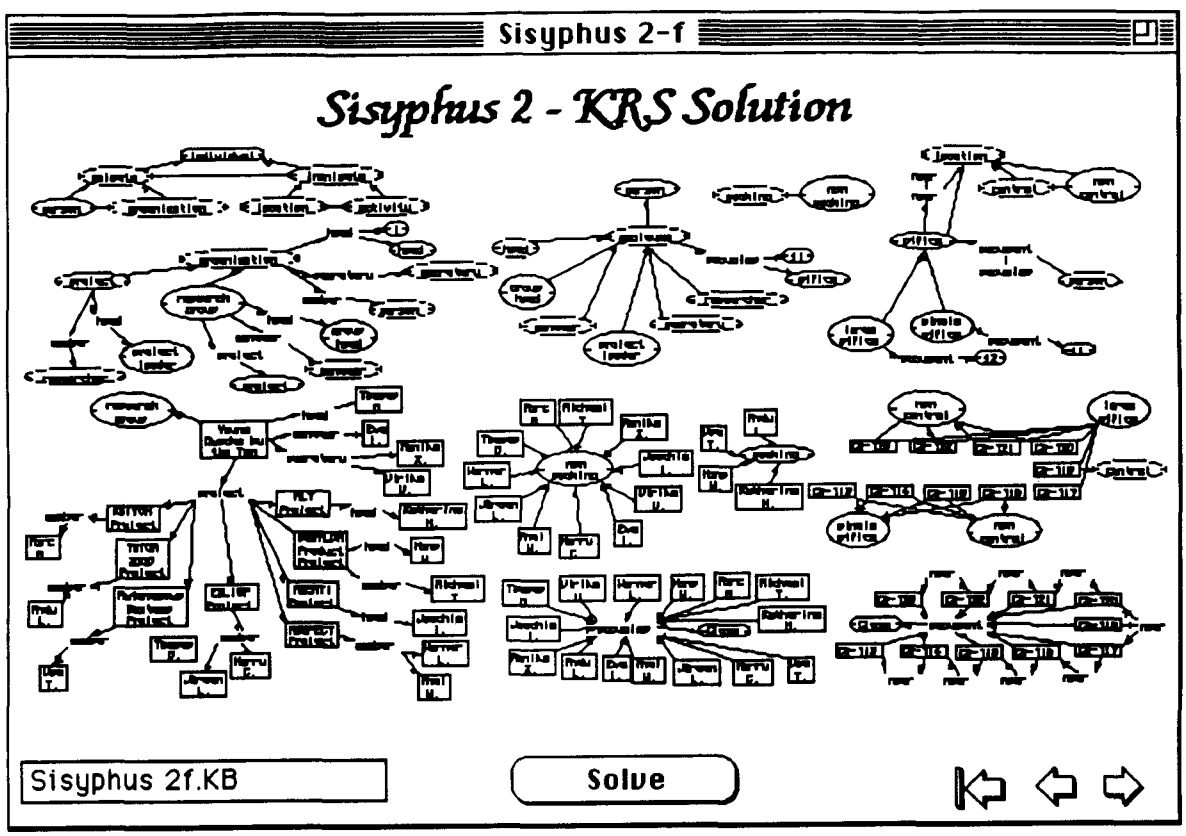

Figure 13 Initial HyperCard screen.

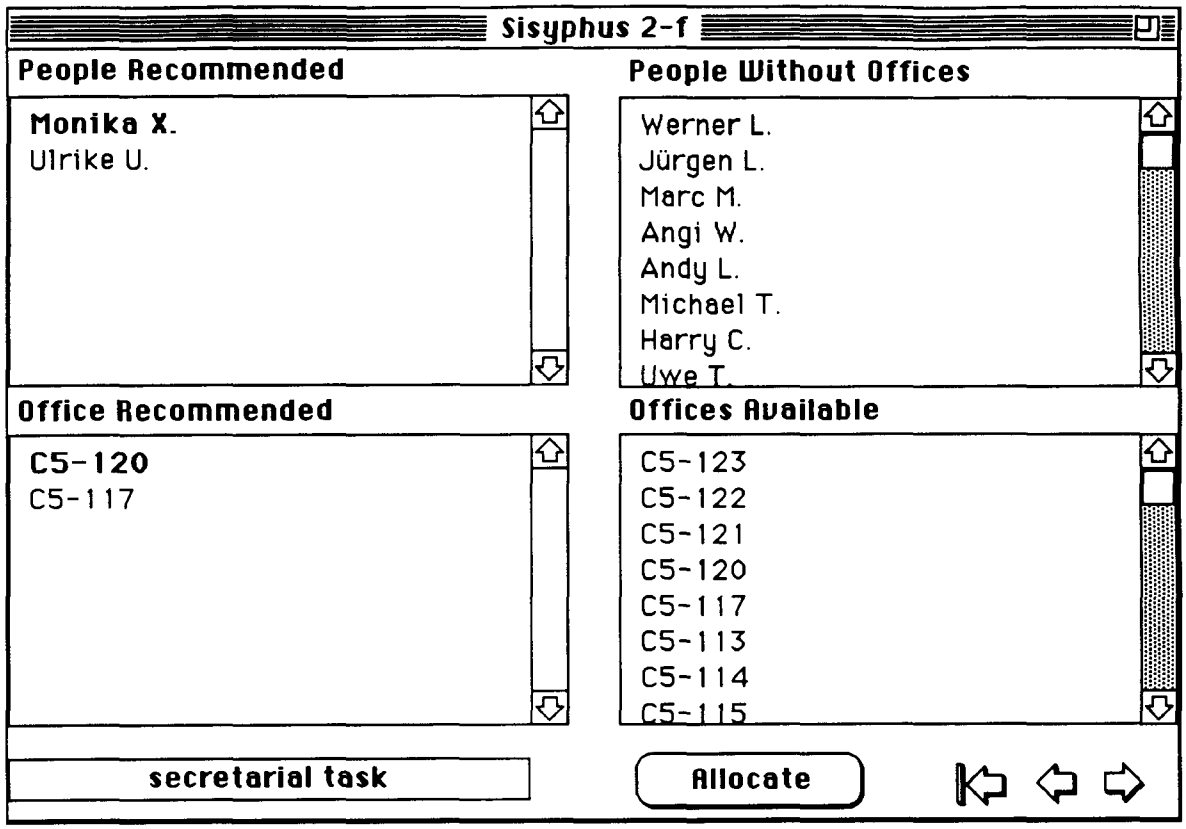

Figure 14 Secretarial room allocation recommendations.

A large office should be allocated to the researchers.

It is not sufficient just to implement these rules. Resource allocation problems tend to be overdetermined, and so notionally insoluble, or underdetermined, and hence subject to combinatorial explosion, and they also readily flip between the two extremes with minor changes to constraints or data. People tend to deal with this by problem reformulation, which is a strongly knowledge-based process, and this reformulation is expedited by information systems that generate meaningful partial solutions that indicate the sources of obstacles to solution. What is required is a system that uses these rules to suggest allocations, allowing the user to make choices when the problem is underdetermined, resolve conflicts when the problem is overdetermined, and supporting retraction and backtracking if the user wishes to explore alternative solutions, perhaps involving considerations not expressed in the knowledge base.

The rules above were used to program KSSn for a generic classification task in which people are classified according to the classes of rooms they need and rooms are classified according to the classes of people they suit. The two classifications are linked through a blackboard mechanism implementing an agenda of tasks ordered by their priority. The classification and agenda mechanisms are programmed visually in KDraw as shown below—neither preexists within the inference engine.

Figure 8 shows the concept definitions for agenda items. At the bottom is a rule for putting actionable items on the "room allocation agenda." An actionable item is defined as one where there is at least one person classified to fill the "person recommended" role in an "office allocation task" and at least one room classified to fill the "room recommended" role in the same task. Figure 9 defines the eight tasks corresponding to the eight rules above.

Figure 10 shows a tacit rule used in determining whether a room is "near" to that of the "group head." The upper left concept defines a "head office" as an "office" with at least one occupant and such that every occupant is a "group head." The rule "near head office" then asserts that an office "near" such an office is "near head office," that is, the con-

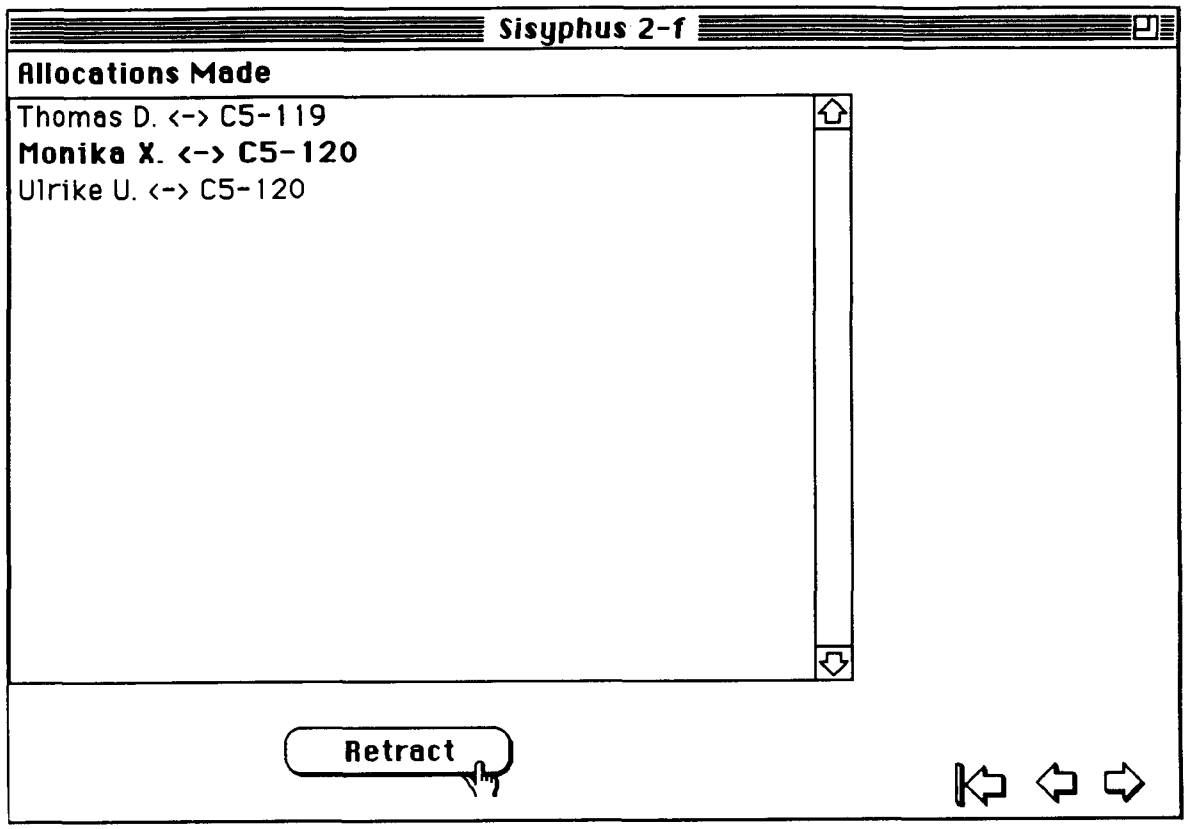

Figure 15 Allocations and retraction.

cept "near head office" will be asserted of it and it may then be recognized as falling under other rules that require this constraint. The lower left concept defines an "occupied near head office" office as one that is "near head office" and has at least one occupant. Such an office again has its neighbors classified as "near head office." This captures the expert's reclassification of offices further and further away from head office as being "near" to it as intervening offices are filled.

Figure 11 shows the classification rules corresponding to the allocation rules above. For example, at the top level if an individual is a group head and has no individuals in the "occupies" role then that individual is classified as "group head without office" and the "group head request" rule fires placing "group head task" in the "person recommended" role of the individual. Hence, because this role is self-inverse the individual is inferred to be in the "person recommended" role of "group head task."

All of the knowledge structures shown in Figure 7 have now been shown in detail. They are the user interface for domain, problem-solving, and particular problem description. It remains to describe the user interface for problem solving. KRS is a server and KDraw a knowledge entry and editing subsystem defining a knowledge base. It is possible to use either textual or graphic querying of KRS to solve a particular problem. However, for the Sisyphus example the use of HyperCard to provide an open-architecture user interface to KRS was demonstrated. Figure 12 shows the overall architecture.

HyperCard communicates with KRS, as does KDraw, through Apple's System 7 interapplication communication protocol. This allows a server instantiation to exist anywhere on a network and be accessed locally or remotely. Figure 13 shows the initial screen in HyperCard. The "knowledge structures" shown on this card are decorative only, not active. HyperCard is being used only as a user interface to query the knowledge base. The user enters the KRS server name containing the knowledge base defined above at the lower left and clicks on the "Solve" button. The user will then see a sequence of recommended room allocations. In general, if the problem is underdetermined this will involve some choice. Figure 14 shows the screen when the group head has been allocated and now the secretarial allocations are possible and top priority.

Clicking on a person and room selects them, and clicking the "Allocate" button causes a message to be sent to KRS allocating the room to the person. The windows on the right in Figure 14 show all the employees without rooms and all the rooms available so that the user can override the recommendation process completely if desired. The user can also go to the screen shown in Figure 15 at any time, see the allocations already made, select any number of them, and retract them if desired. The KRS truth maintenance system automatically undoes any conclusions based on retracted data.

CONCLUSIONS

Object-oriented databases and knowledge representation, and visual languages, are being developed and applied by a number of research groups as organizational information technologies. This article has reported some experience in the design and implementation of a lightweight, object-oriented knowledge representation server and its application to organizational modeling and problem solving. The primary user interface is through a formal visual language implemented simply and naturally as a drawing environment on graphic workstations. The open-architecture implementation of the server allows it to be integrated with existing applications, such as corporate database and accounting systems, and also allows additional functionality to be added through self-contained modules requiring no changes in the kernal system.

The problem-solving example given has shown how knowledge entered visually can be used to model organizational structures in a way that is simple and natural and leads directly to operational problem solving. The process described here has the following features:

- The knowledge and data structures are totally overt and easily edited.
- The visual language allows knowledge associated with structures such as the organization chart and room layout to be presented naturally.
- The problem-solving strategy is incremental and can be applied to an existing configuration of rooms and occupancies.
- The arbitrary choices that arise in undetermined problems can be made by the system or made by a person with, perhaps, additional considerations in mind.
- Condensed and understandable information is available at the interface (the agenda items, consisting of pairs of sets of people and rooms, are amenable to attractive presentation).
- The solution developed is a highly generic problem-solving strategy.
- The various components of the solution may be envisioned as coming from different archives in a corporate knowledge repository.

These features exemplify the rationale for the approach taken to organizational modeling and problem solving described in the first part of this article.

ACKNOWLEDGMENTS

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada. The author is grateful to many colleagues for discussions over the years that have influenced the research described in this article, particularly Mildred Shaw for research collaboration that has shaped many of the ideas reported in this article and Marc Linster, the GMD, and the Esprit *Reflect* Project for making this dataset available through Sisyphus.

REFERENCES

Abrett, G. and Burstein, M. H. (1988) "The KREME Knowledge Editing Environment," In J. H. Boose and B. R. Gaines, Eds., *Knowledge Acquisition Tools for Expert Systems.* Academic Press, London. pp. 1–24.

Afsarmanesh, H. and McLeod, D. (1989) "The 3DIS: An Extensible Object-Oriented Information Management Environment," *ACM Trans. Info. Systems* 7:339–377.

Borgida, A., Brachman, R. J., McGuiness, D. L., and Resnick, L. A. (1989) "CLASSIC: A Structural Data Model for Objects," *Proc. 1989 SIGMOD Conf. Manag. Data,* pp. 58–67.

Brachman, R. J. (1977) "What's In a Concept: Structural Foundations for Semantic Nets," *Int. J. Man-Mach. Stud.* 9:127–152.

Brachman, R. J. "On the Epistemological Status of Semantic Nets," In N. V. Findler, Ed., *Associative Networks: Representation and Use of Knowledge by Computers.* Academic Press, New York, pp. 3–50.

Brachman, R. J., Gilbert, V. P., and Levesque, H. J. (1985) "An Essential Hybrid Reasoning System: Knowledge and Symbol Level Accounts of KRYPTON," *Proc. IJCAI85,* pp. 547–551.

Brachman, R. J. and Levesque, H. J. (1984) "The Tractability of Subsumption in Frame-Based Description Languages," *Proc. AAAI-84,* pp. 34–37.

Brachman, R. J. and Schmolze, J. (1985) "An Overview of the KL-ONE Knowledge Representation System," *Cogn. Sci.* 9:171–216.

Cercone, N. and Schubert, L. (1975) "Towards a State-Based Conceptual Representation," *Proc. AAAI75,* pp. 83–90.

Davis, J. P. (1989) The Knowledge Dictionary: A KBMS Architecture for the Many-to-Many Coupling of Knowledge Based-Systems to Databases, University of South Carolina.

Fahlman, S. E. (1979) *NETL: A System for Representing and Using Real-World Knowledge.* MIT Press, Cambridge, MA.

Gaines, B. R. (1988a) "Knowledge Acquisition Systems for Rapid Prototyping of Expert Systems," *INFOR* 26: 256–285.

Gaines, B. R. (1988b) "Structure, Development and Applications of Expert Systems in Integrated Manufacturing," In A. Kusiak, *Artificial Intelligence Implications for CIM.* Ed., IFS Conferences, Bedford, UK, pp. 117–161.

Gaines, B. R. (1991a) "Empirical Investigations of Knowledge Representation Servers: Design Issues and Applications Experience with KRS," *ACM SIGART Bull.* 2(3):45–56.

Gaines, B. R. (1991b) "Integrating Rules in Term Subsumption Knowledge Representation Servers," *AAAI'91: Proc. 9th National Conf. Artif. Intell.*, pp. 458–463.

Gaines, B. R. (1991c) "An Interactive Visual Language for Term Subsumption Visual Languages," *IJCAI'91: Proc. 12th Int. Joint Conf. Artif. Intell.*, pp. 817–823.

Gaines, B. R., Rappaport, A., and Shaw, M. L. G. (1989) "A Heterogeneous Knowledge Support System," *Proc. 4th AAAI Knowledge Acquisition Knowledge-Based Systems Workshop,* pp. 13-1–13-20.

Gaines, B. R. and Shaw, M. L. G. (1990) "Cognitive and Logical Foundations of Knowledge Acquisition," *Proc. 5th AAAI Knowledge Acquisition Knowledge-Based Systems Workshop,* pp. 9-1–9-24.

Glinert, I. P., Ed. (1990) *Visual Programming Environments: Paradigms and Systems.* IEEE Computer Society Press, Los Alamitos, CA.

Gupta, R. (1991) *Object-Oriented Databases with Applications to CASE, Networks, and VLSI CAD.* Prentice Hall, Englewood Cliffs, NJ.

Güting, R. H., Zicari, R., and Choy, D. M. (1989) "An Algebra for Structured Office Documents," *ACM Trans. Info. Systems* 7:123–157.

Kindermann, C. and Quantz, J. (1989) *Graphics-Oriented User Interfaces for KL-ONE,* KIT Internal Report 23, Technical University of Berlin.

Lee, J. and Malone, T. W. (1990) "Partially Shared Views: A Scheme for Communicating Among Groups That Use Different Type Hierarchies," *ACM Trans. Info. Systems* 8:1–26.

Levesque, H. J. (1984) "A Logic of Implicit and Explicit Belief." *Proc. AAAI-84,* pp. 198–202.

Linster, M. (1991a) *Sisyphus Working Papers Part 2: Models of Problem Solving.* GMD, Bonn, Germany.

Linster, M., Ed. (1991b) *Sisyphus Working Papers Part 2: Models of Problem Solving.* GMD, Bonn, Germany.

MacGregor, R. M. (1988) "A Deductive Pattern Matcher," *Proc. AAAI88,* pp. 403–408.

Mackinlay, J. and Genesereth, M. R. (1984) "Expressiveness of Languages," *Proc. AAAI84,* pp. 226–232.

Nebel, B. (1990) *Reasoning and Revision in Hybrid Representation Systems,* vol. LNCS-422. Springer, Berlin.

Nosek, J. T. and Roth, I. (1990) "A Comparison of Formal Knowledge Representations as Communication Tools: Predicate Logic vs Semantic Network," *Int. J. Man–Mach. Stud.* 33:227–239.

Parsaye, K., Chignell, M., Khoshafian, S., and Wong, H. (1989) *Intelligent Databases: Object-Oriented, Deductive Hypermedia Technologies.* Wiley, New York.

Pernici, B., Barbic, F., Fugini, M. G., Maiocchi, R., Rames, J. R., and Rolland, C. (1989) "C-TODOS: An Automatic Tool for Office System Conceptual Design," *ACM Trans. Info. Systems* 7:378–419.

Pernici, B. and Rolland, C., Eds. (1990) *Automatic Tools for Designing Office Information Systems.* Springer, Berlin.

Quillian, M. R. (1968) "Semantic Memory," in M. Minsky, Ed., *Semantic Information Processing.* MIT Press, Cambridge, MA, pp. 216–270.

Runes, D. D. (1972) *Dictionary of Philosophy.* Peter Owen, London.

Sagawa, J. M. (1990) "Repository Manager Technology," *IBM J. Res. Dev.* 29:209–226.

Scheer, A.-W. (1989) *Enterprise-Wide Data Modeling.* Springer, Berlin.

Schmidt-Schauβ, M. (1989) "Subsumption in KL-ONE is Undecidable," *Proc. KR'89: 1st Int. Conf. Principles Knowledge Representation and Reasoning,* pp. 421–431.

Schmolze, J. (1983) KLONEDRAW—A Facility for Automatically Drawing Pictures of KL-ONE Networks," In *Research in Knowledge Representation for Natural Language Understanding. Report 5421.* Bolt Beranek and Newman, Cambridge, MA, pp. 41–44.

Sciore, E. (1989) "Object Specialization," *ACM Trans. Info. Systems* 7:103–122.

Scott, W. R. (1981) *Organizations: Rational, Natural and Open Systems.* Prentice Hall, Englewood Cliffs, NJ.

Shaw, M. L. G. and Gaines, B. R. (1987) "KITTEN: Knowledge Initiation and Transfer Tools for Experts and Novices," *Int. J. Man–Mach. Stud.* 27:251–280.

Shriver, B. and Wegner, P. (1987) *Research Directions in Object-Oriented Programming.* MIT Press, Cambridge, MA.

Voβ, A., Karbach, W., Drouven, U., Lorek, D., and Schuckey, R. (1990) "Operationalization of a Synthetic Problem," In *ESPRIT Basic Research Project P3178 REFLECT Task I.2.1 Report.* GMD, Bonn, Germany.

Watanabe, H. (1989) "Heuristic Graph Displayer for G-BASE," *Int. J. Man-Mach. Stud.* 30:287–302.

Woods, W. A. (1975) "What's In a Link: Foundations for Semantic Networks," In D. G. Bobrow and A. M. Collins, Eds. *Representation and Understanding: Studies in Cognitive Science,* Academic Press, New York, pp. 35–82.