ORGANIZATIONAL MODELING AND PROBLEM SOLVING USING AN OBJECT-ORIENTED KNOWLEDGE REPRESENTATION SERVER AND VISUAL LANGUAGE

Brian R Gaines
Knowledge Science Institute, University of Calgary
Calgary, Alberta, Canada T2N 1N4. gaines@cpsc.ucalgary.ca

Object-oriented knowledge representation systems allow an organization to be modeled in an understandable way. User interaction with such a system is enhanced if the graphic representations normally used, such as organization charts and room layout plans, are supported through visual languages directly generating the underlying knowledge structures. The operational and problem-solving procedures of an organization expressed visually within this framework become overt and accessible, and their implementation and consequences can be automated and evaluated. This paper reports 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 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 functionality to be added through self-contained modules requiring no changes in the kernel system.

1 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 complex structures have been only implicitly represented in the information technology developed to support organizations. 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, 1988a). 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 (Sciore, 1989; Afsarmanesh & McLeod, 1989), and to significant sub-systems such as documents (Güting, Zicari & Choy, 1989) and electronic mail (Lee & Malone, 1990). In particular, the ESPRIT TODOS project for the design of office information systems has adopted object-oriented design through conceptual modeling on a large-scale (Pernici, Barbic, Fugini, Maiocchi, Rames & Rolland, 1989; Pernici & Rolland, 1990). The routine application of the object-oriented modeling approach is yet to come, although expert system shells such as ART, KEE and NEXPERT all provide object-oriented knowledge representation environments, and there is convergence between artificial intelligence, hypermedia and knowledge base technologies (Gupta, 1991; Parsaye, Chignell, Khohafian & Wong, 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 since 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).

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
© 1991 ACM 0-89791-456-2/91/0010/0080...$1.50

This paper reports some studies of organizational modeling and problem solving based on the hypothesis that it is the user interface to corporate knowledge bases which 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 makes 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 light-weight knowledge representation server, and to use the object-oriented programming paradigm to surround the server with small functional modules supporting a variety of applications (Gaines, 1991a,b). The server is sufficiently light-weight 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.

The following sections outline the system architecture and knowledge representation server, the formal visual language for knowledge representation, and an example application to room allocation in a research organization.

2 KSSn ARCHITECTURE

KSSn (Knowledge Support System n) is the latest in a series of developments deriving from our initial implementation of KSS0 (Gaines, 1988b) and KSS1 (Shaw & 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 & Shaw, 1989; Gaines & Linster, 1990). One problem with research and applications in this area is that of rapid development in techniques, tools and technologies. Systems become obsolescent and a monolithic architecture is difficult to maintain and enhance. To cope with this we have designed 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 insights into the complexity issues affecting knowledge representation and reasoning (Brachman & Levesque, 1984; Schmidt-Schauss, 1989); and the development of object-oriented programming environments offering precise definition and efficient implementation of extensible abstract data types (Shriver & 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, Brachman, McGuinness & Resnick, 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 & Schmolze, 1985), developing through KRYPTON (Brachman, Gilbert & Levesque, 1985) and currently culminating in systems such as CLASSIC (Borgida, Brachman, McGuinness & Resnick, 1989) and LOOM (MacGregor, 1988) has reached a maturity of technology which 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 & Levesque, 1984; Schmidt-Schauss, 1989; Nebel, 1990). 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.

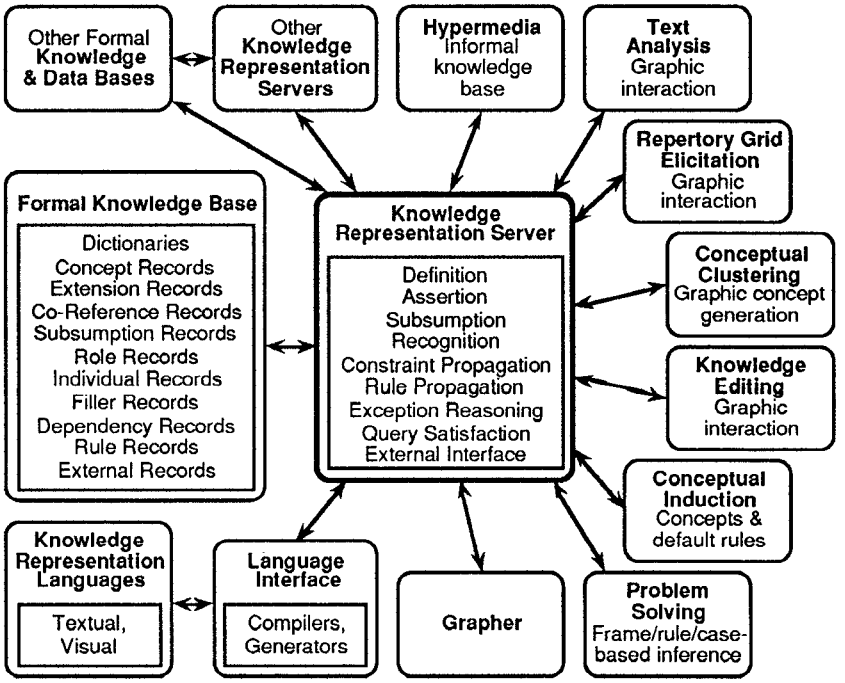

Fig.1 Architecture of the knowledge representation server

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 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 and dates, and with rule representation that allows one rule to be declared an exception to others. For the purposes of this paper 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.

3 KDRAW VISUAL LANGUAGE

An important component of KSSn in the context of this paper is the graphic knowledge editor, 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) have 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 learnt and understood (Gaines & 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 (Woods, 1975; Brachman, 1977). Nodes, arcs and their labels could be used very freely and ambiguously and diagrams were subject to differing interpretations. In the 1970s there were proposals for network formalisms with well-defined semantics (Cercone & Schubert, 1975; Fahlman, 1979; Brachman, 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 (Borgida *et al*, 1989).

In the light of these developments semantic networks were re-examined 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 (that is, translate unambiguously in both directions) between these languages and knowledge representation formalisms with well-defined semantics?

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 & 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.

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 structure 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.

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 are ovals with small horizontal lines inside each side, individuals rectangles, roles unboxed text, rules rounded-corner boxes, 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, e.g.:

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

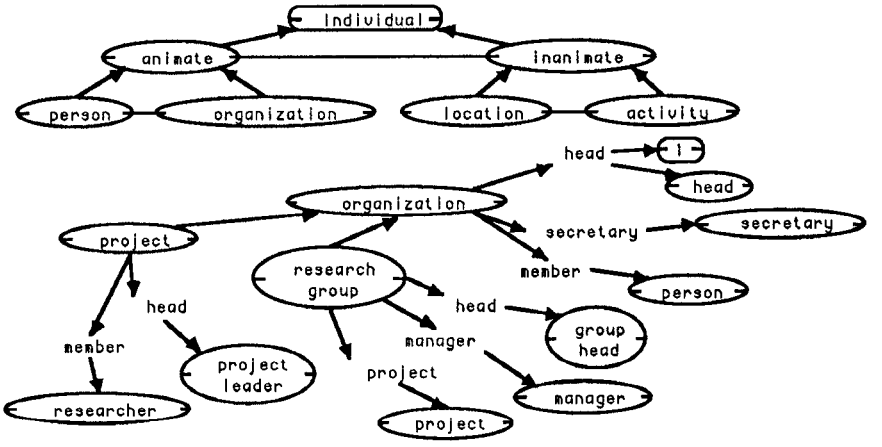

Fig.2 Top-level and organizational ontologies in KDraw visual language

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" (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, since it is shown to inherit from "organization" also has these roles and constraints but is further constrained to have its "head" role filled by a "group head" and its "member" role filled by individuals of type "researcher".

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."

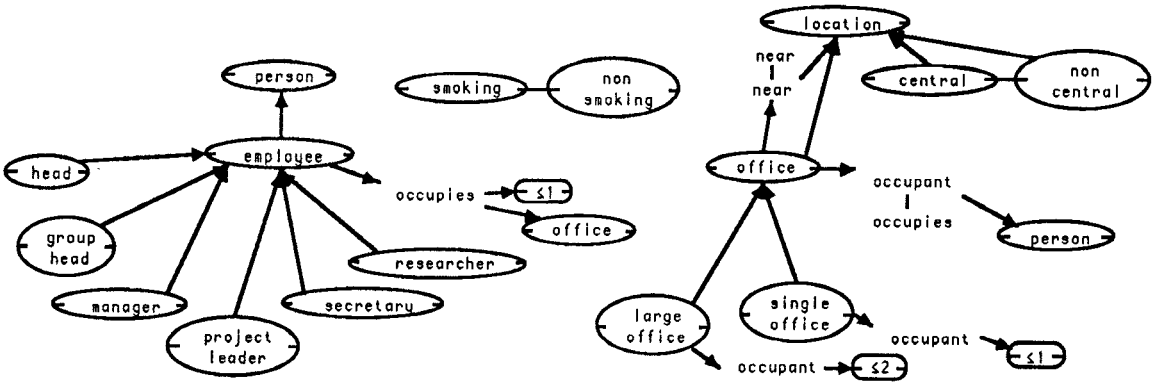

Fig.3 Employee and office ontologies

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 "YQT Research Center" 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 "YQT Research Center".

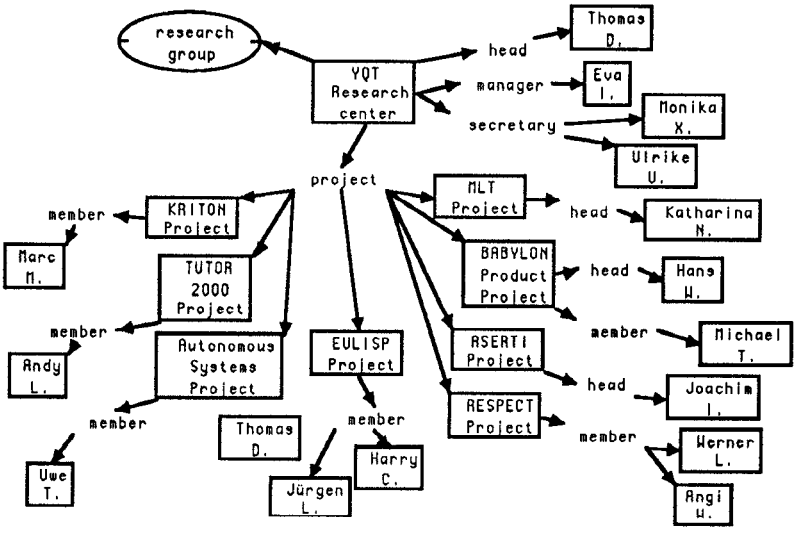

Fig.4 Facts asserted about projects and employees as an organization chart

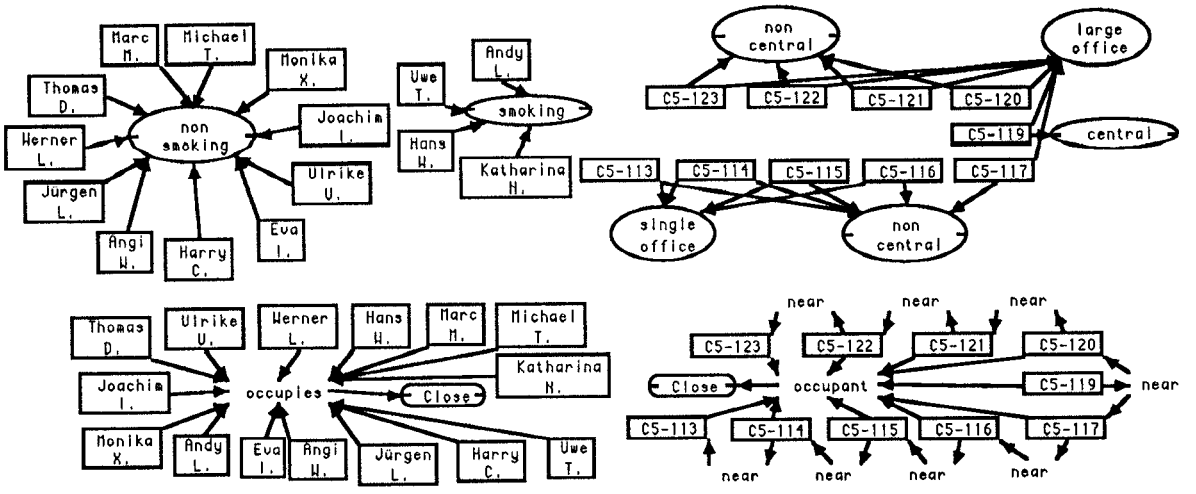

Fig.5 Facts asserted about employees and about rooms as a room layout plan

Figure 5 supplies 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 papers 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 cosmetic issues like layout. 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 5 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 right of Figure 5, 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 6 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. However, the structures shown in full in Figures 2 through 5 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 6 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 the minimum of effort. Updates are efficient and graphs with over a thousand 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, 1991c). The following example extends that described above to illustrate these features.

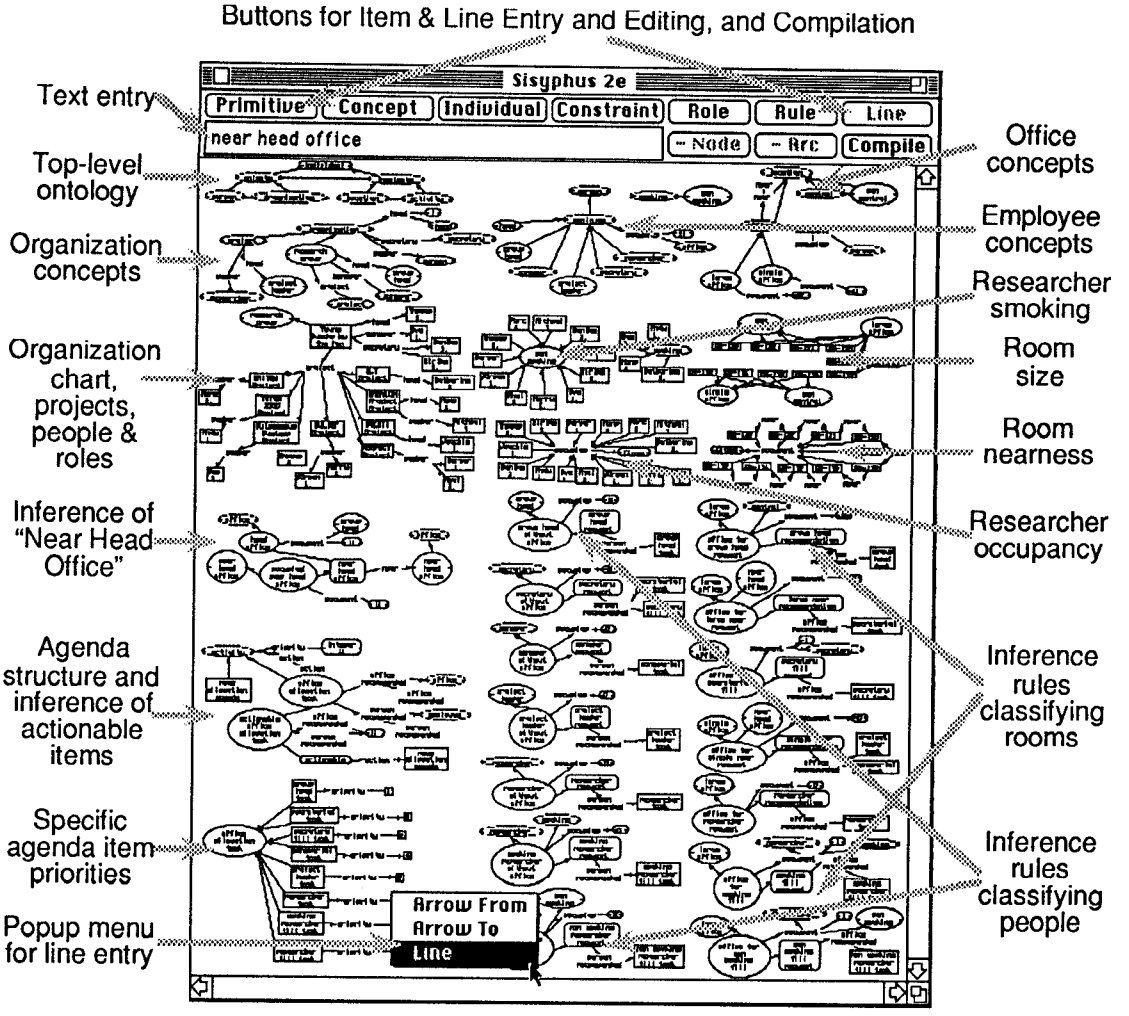

Fig.6 Problem, its solution and example dataset expressed in KDraw visual language

The problem is one of room allocation from an ESPRIT project (Voß, Karbach, Drouven, Lorek & Schuckey, 1990) that has recently been made part of Project Sisyphus. Sisyphus is research 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, 1991). Some sixteen research groups internationally participated and the example 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 paper, 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 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 non-smoking researcher occupant should be allocated to another non-smoking researcher.
A large office should be allocated to the researchers.

It is not sufficient just to implement these rules. Resource allocation problems tend to be over-determined, and so notionally insoluble, or under-determined, 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 under-determined, resolve conflicts when the problem is over-determined, and supporting retraction and backtracking if the user wishes to explore alternative solutions, perhaps involving considerations not expressed in the knowledge base.

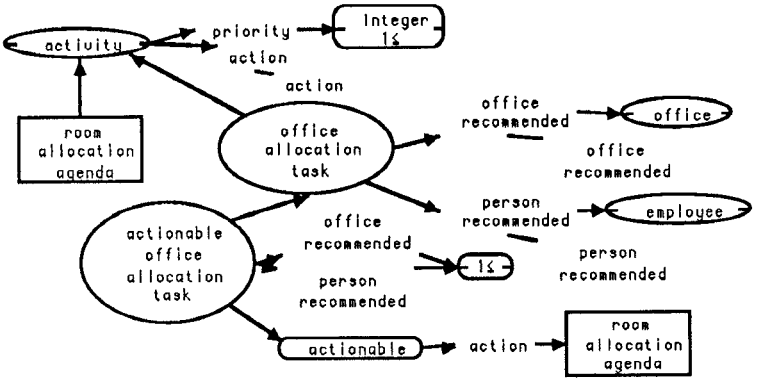

Fig.7 Concept definitions for agenda mechanism and rule putting items on it

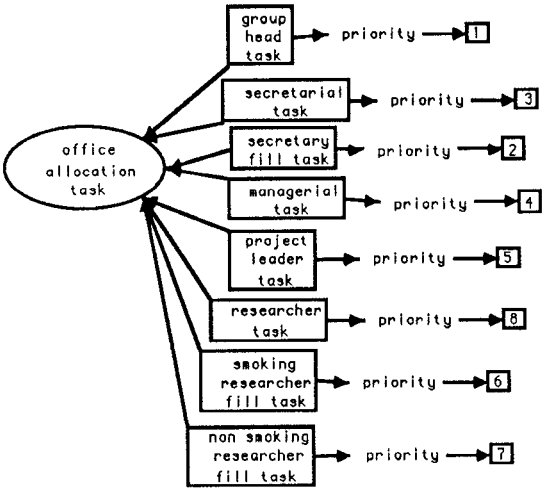

Fig.8 Priorities asserted for possible agenda items

Figure 9 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 concept "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 which 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.

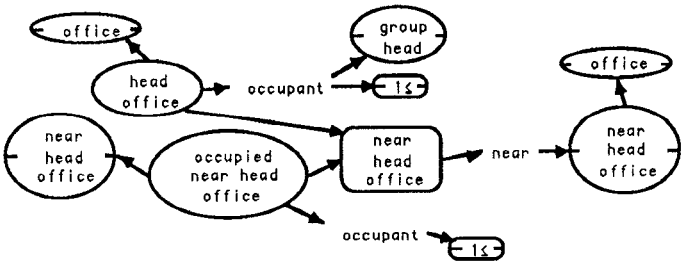

Fig.9 Rule inferring "near head office" and propagating it as offices fill

Figure 10 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, since this role is self-inverse, the individual is inferred to be in the "person recommended" role of "group head task".

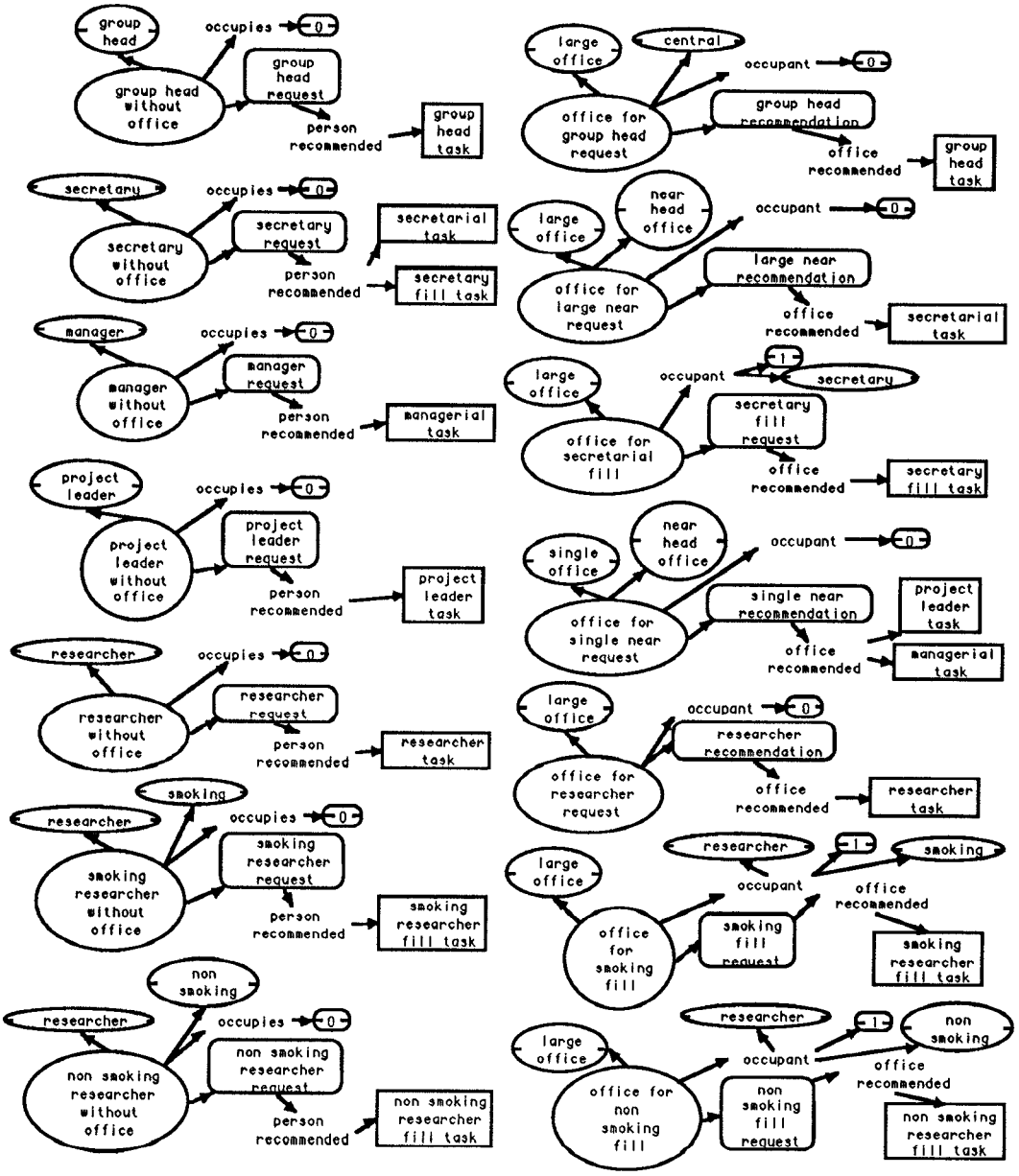

Fig.10 Rules classifying people (on left) and offices (on right)

All of the knowledge structures shown in Figure 6 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 is a knowledge entry and editing sub-system 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 11 shows the overall architecture.

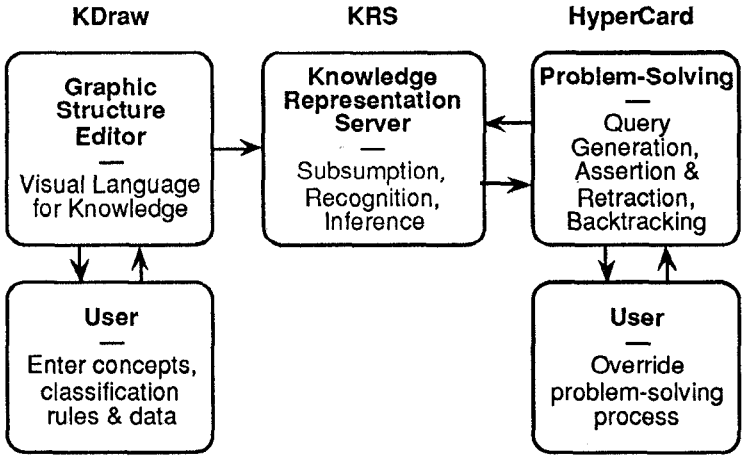

Fig.11 System architecture: KSSn sub-systems, functions and user interaction

HyperCard communicates with KRS, as does KDraw, through Apple's System 7 inter-application communication protocol. This allows a server instantiation to exist anywhere on a network and be accessed locally or remotely. Figure 12 shows the initial screen in HyperCard. The user has only to enter the KRS server name containing the knowledge base defined above at the lower right and click on the "Solve" button. The user will then see a sequence of recommended room allocations. In general, if the problem is under-determined, this will involve some choice. Figure 13 shows the screen when the group head has been allocated and now the secretarial allocations are possible and top priority.

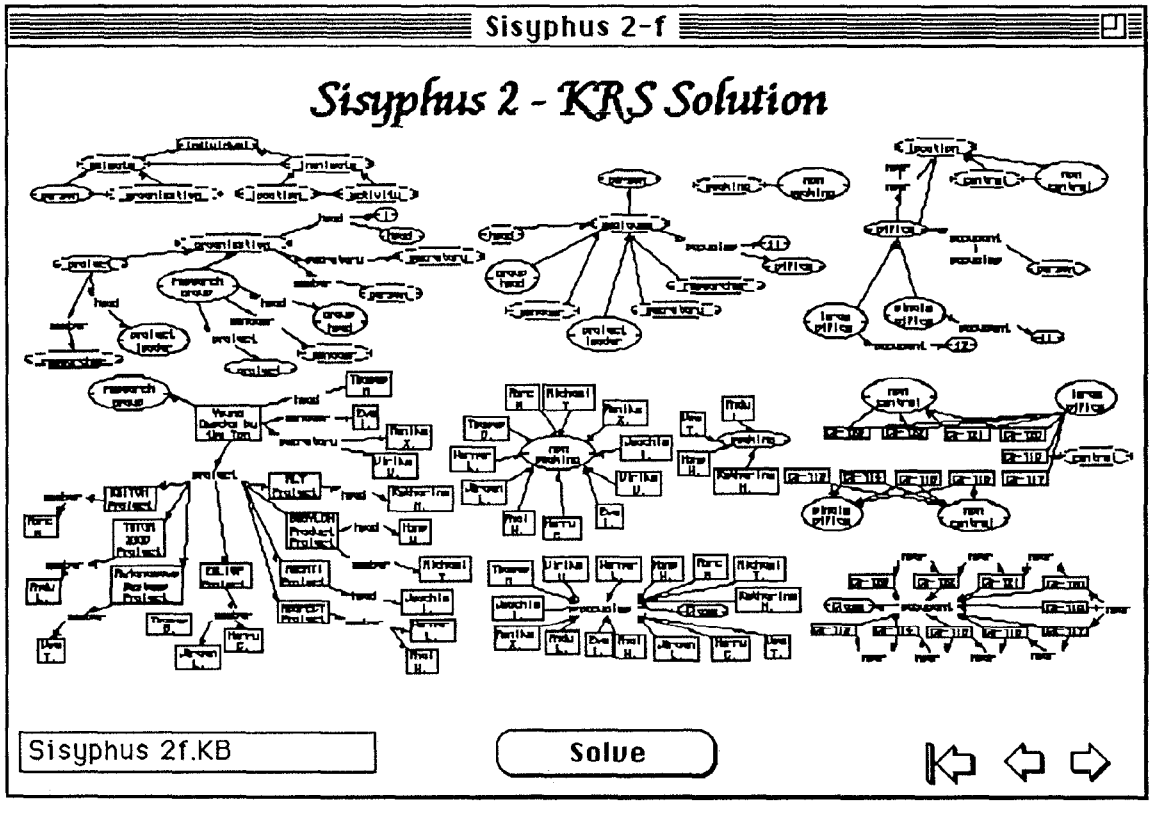

Fig.12 Initial HyperCard screen

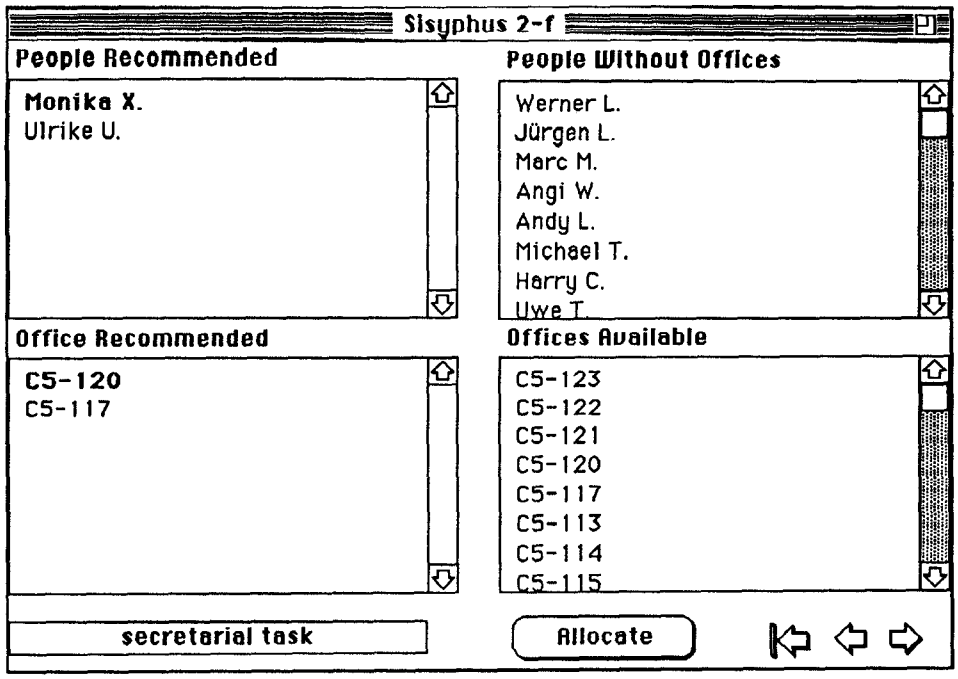

Fig.13 Secretarial room allocation recommendations

Clicking on a person and room selects them, and clicking the "Allocate" button causes a message to be send to KRS allocating the room to the person. The windows on the right in Figure 13 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 14 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.

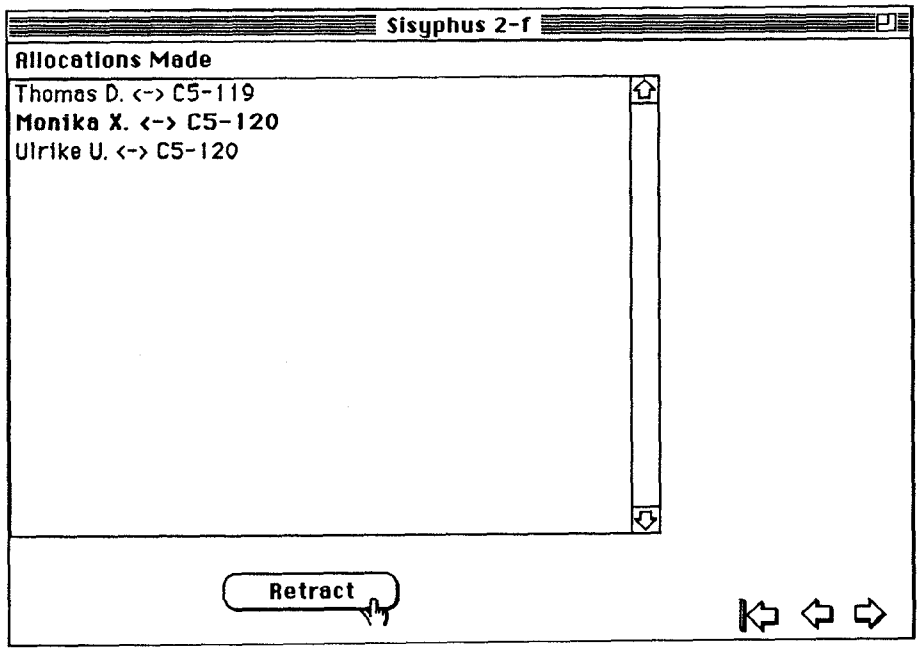

Fig.14 Allocations and retraction

4 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 paper 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 kernel 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 very 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 paper.

ACKNOWLEDGEMENTS

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

REFERENCES

Abrett, G. & Burstein, M.H. (1988) The KREME knowledge editing environment. In Boose, J.H. & Gaines, B.R., Eds. **Knowledge Acquisition Tools for Expert Systems.** pp.1-24. London, Academic Press.

Afsarmanesh, H. & McLeod, D. (1989). The 3DIS: an extensible object-oriented information management environment. **ACM Transactions on Information Systems 7**(4) 339-377 (October).

Borgida, A., Brachman, R.J., McGuiness, D.L. & Resnick, L.A. (1989). CLASSIC: a structural data model for objects. Clifford, J., Lindsay, B. & Maier, D., Eds. **Proceedings of 1989 ACM SIGMOD International Conference on the Management of Data.** pp.58-67. New York: ACM Press.

Brachman, R.J. (1977) What's in a concept: structural fondations for semantic nets. **International Journal of Man-Machine Studies 9**, 127-152.

Brachman, R.J. (1979). On the epistemological status of semantic nets. In Findler, N.V., Ed. **Associative Networks: Representation and Use of Knowledge by Computers.** pp.3-50. New York: Academic Press.

Brachman, R. J., Gilbert, V.P. & Levesque, H. J. (1985). An essential hybrid reasoning system: knowledge and symbol level accounts of KRYPTON. **Proceedings of IJCAI85.** pp.547-551. Los Altos: Morgan Kaufmann.

Brachman, R. J., & Levesque, H. J. (1984) The tractability of subsumption in frame-based description languages. **Proceedings of AAAI-84.** pp.34-37. San Mateo, California: Morgan Kaufmann.

Brachman, R.J. & Schmolze, J. (1985). An overview of the KL-ONE knowledge representation system. **Cognitive Science, 9**(2) 171-216 (April-June).

Cercone, N. & Schubert, L. (1975). Towards a state-based conceptual representation. **Proceedings of AAAI75.** pp.83-90. Los Altos: Morgan Kaufmann.

Davis, J.P. (1989). **The Knowledge Dictionary: A KBMS Architecture for the Many-to-Many Coupling of Knowledge Based-Systems to Databases.** PhD Thesis, University of South Carolina, UMI9005097

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

Gaines, B.R. (1988a). Structure, development and applications of expert systems in integrated manufacturing. Kusiak, A., Ed. **Artificial Intelligence Implications for CIM.** pp.117-161. Bedford, UK: IFS Conferences.

Gaines, B.R. (1988b). Knowledge acquisition systems for rapid prototyping of expert systems. **INFOR, 26**(4), 256-285 (November).

Gaines, B.R. (1991a) Empirical investigation of knowledge representation servers: design issues and applications experience with KRS. **AAAI Spring Symposium: Implemented Knowledge Representation and Reasoning Systems.** pp. 87-101. Stanford (March)—**SIGART Bulletin 2**(3), 45-56 (June).

Gaines, B.R. (1991b). Integrating rules in term subsumption knowledge representation servers. **AAAI'91: Proceedings of the Ninth National Conference on Artificial Intelligence.** pp.458-463. Menlo Park, California: AAAI Press/MIT Press (July).

Gaines, B.R. (1991c). An interactive visual language for term subsumption visual languages. **IJCAI'91: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence.** San Mateo, California: Morgan Kaufmann.

Gaines, B.R. & Linster, M. (1990). Integrating a knowledge acquisition tool, an expert system shell and a hypermedia system. **International Journal of Expert Systems Research and Applications 3**(2) 105-129.

Gaines, B.R., Rappaport, A. & Shaw, M.L.G. (1989). A heterogeneous knowledge support system. Boose, J.H. & Gaines, B.R., Eds. **Proceedings of the Fourth AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop.** pp.13-1-13-20. Banff (October).

Gaines, B.R. & Shaw, M.L.G. (1990) Cognitive and logical foundations of knowledge acquisition. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Fifth AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop.** pp. 9-1-9-25. Banff (November).

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

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

Güting, R.H., Zicari, R. & Choy, D.M. (1989). An algebra for structured office documents. **ACM Transactions on Information Systems** 7(2) 123-157 (April).

Kindermann, C. & Quantz, J. (1989) **Graphics-oriented user interfaces for KL-ONE.** KIT Internal Report 23. Technical University of Berlin.

Lee, J & Malone, T.W. (1990). Partially shared views: a scheme for communicating among groups that use different type hierarchies. **ACM Transactions on Information Systems 8**(1) 1-26 (January).

Levesque, H. J. (1984) A logic of implicit and explicit belief. **Proceedings of AAAI-84.** pp.198-202. San Mateo, California: Morgan Kaufmann.

Linster, M., Ed. (1991) **Sisyphus Working Papers Part 2: Models of Problem Solving** EKAW91, Glasgow: University of Strathclyde.

MacGregor, R.M. (1988) A deductive pattern matcher. **Proceedings of AAAI88.** pp.403-408. San Mateo: Morgan Kaufmann.

Nebel, B. (1990) **Reasoning and Revision in Hybrid Representation Systems.** Berlin: Springer-Verlag.

Nosek, J.T. & Roth, I. (1990) A comparison of formal knowledge representations as communication tools: predicate logic vs semantic network. International Journal of Man-Machine Studies 33, 227-239, 1990.

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

Pernici, B., Barbic, F., Fugini, M.G., Maiocchi, R., Rames, J.R. & Rolland, C. (1989). C-TODOS: an automatic tool for office system conceptual design. **ACM Transactions on Information Systems 7**(4) 378-419 (October).

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

Quillian, M.R. (1968). Semantic memory. Minsky, M, Ed. **Semantic Information Processing.** pp.216-270. Cambridge, Massachusetts: MIT Press.

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

Sagawa, J.M. (1990). Repository manager technology. **IBM Journal of Research and Development 29**(2) 209-226.

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

Schmidt-Schauss, M. (1989) Subsumption in KL-ONE is undecidable **Proceedings of KR'89: First International Conference on Principles of Knowledge Representation and Reasoning.** pp.421-431. San Mateo, California: Morgan Kaufmann (May).

Schmolze, J. (1983). KLONEDRAW—a facility for automatically drawing pictures of KL-ONE networks. **Research in Knowledge Representation for Natural Language Understanding.** pp.41-44. Report No.5421, Cambridge, Massachusetts: Bolt Beranek and Newman Inc.

Sciore, E. (1989). Object specialization. **ACM Transactions on Information Systems 7**(2) 103-122 (April).

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

Shaw, M.L.G. & Gaines, B.R. (1987). KITTEN: Knowledge Initiation & Transfer Tools for Experts & Novices. **International Journal of Man-Machine Studies, 27,** 251-280.

Shriver, B. & Wegner, P. (1987) **Research Directions in Object-Oriented Programming.** Massachusetts: MIT Press.

Voß, A., Karbach, W., Drouven, U., Lorek, D. & Schuckey, R. (1990) Operationalization of a synthetic problem. **ESPRIT Basic Research Project P3178 REFLECT Task I.2.1 Report** (July).

Watanabe, H. (1989). Heuristic graph displayer for G-BASE. **International Journal of Man-Machine Studies, 30**(3) 287-302 (March).

Woods, W.A. (1975) What's in a link: Foundations for semantic networks. Bobrow, D.G. & Collins, A.M. (Eds) **Representation and Understanding: Studies in Cognitive Science.** pp.35-82. New York: Academic Press.