THE SISYPHUS PROBLEM SOLVING EXAMPLE THROUGH A VISUAL LANGUAGE WITH KL-ONE-LIKE KNOWLEDGE REPRESENTATION

Brian R Gaines

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

ABSTRACT

The Sisyphus room allocation problem solving example has been used as a test of the interactive graphic editor in KSSn as a knowledge entry and editing system, and of the KL-ONE-like language of the knowledge representation server, KRS, that is driven by the visual language of the graphic editor. A *situated action* solution was developed from the dataset provided in terms of a pair of heuristic classification systems, one classifying people and the other rooms, coupled to an agenda mechanism that generates the problem-solving decision sequence.

INTRODUCTION

The Sisyphus knowledge acquisition test cases are a particular challenge to those who have been claiming to have generic, general-purpose knowledge acquisition tools. It is reasonable to expect that within the toolkit there exists something that can tackle each test case, and the question should be how well it does so. The room allocation example is a particular challenge for the KSSn family of tools because it has a major component of sequential problem-solving whereas KSS0 (Gaines & Shaw, 1991) is clearly a toolkit for heuristic classification problems. The most recent extensions of KSSn are towards more complex knowledge representation while attempting to retain the simplicity in the user interfaces of KSS0 and principled, uniform mathematical data structures. However, the type of representation chosen, a term subsumption language modeled on CLASSIC (Borgida, Brachman, McGuiness, & Resnick, 1989), itself modeled on KL-ONE (Brachman & Schmolze, 1985), still has classification as its primary inference process. In the context of knowledge acquisition tools there has been a debate about whether classification lies at the heart of all problem solving methodologies, and Bradshaw, Boose, Covington and Russo (1988) have given examples of its applicability to a range of problems that do not seem classificatory in nature. The room allocation example is an opportunity for those of us who emphasize the universality of classification to solve, or fail to solve, the problem with our existing tools.

The room allocation problem looks as if it has little to do with knowledge-based systems. It is the type of task analyzed in the operations research literature rather than that of artificial intelligence. The ESPRIT report on GMD's analysis discusses this issue (Voß, Karbach, Drouven, Lorek & Schuckey, 1990), and makes a case for exploring the role of knowledge-based systems to resource allocation problems. Such 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 would be expedited by information systems that generated meaningful partial solutions that indicate the sources of obstacles to solution. What characterizes an attractive problem solver for this class of problem may be difficult to determine at this stage, but one outcome of the pluralistic Sisyphus studies should be a much better understanding of the issues.

KSSn APPROACHES

The dataset consists of a protocol of a single instance of an expert solving a problem. This may appear inadequate, but it is interesting to find out how far one can go with this data. Its paucity does reflect a common situation in practice. Research in explanation-based learning (Ellman, 1989) shows that a single example together with a strong domain theory allows significant generalizations. The domain theory of offices and organizations at the level required for this problem is well-defined, and the protocol provides a commentary that extends the domain theory—it is not just a case description. Hence, the problem analysis presented keeps track of the domain theory, the case history, the expert's comments, and possible generalizations.

In terms of tools, this data is already beyond the point where KSS0's repertory grid tools would be useful. If the expert were not reflective and had not already given a clear exposition of the relevant attributes then these tools would be useful. That is, given an exemplary set of people how do they differ on relevant dimensions, and similarly given an exemplary set of rooms. A text clustering analysis of a discussion with the expert would be useful in initiating the repertory grid tool with a relevant vocabulary. One would expect the grid to elicit job classes, seniorities, smoking habits, equipment requirements, and so on for people, and size, location, resources, and so on for rooms. However, these dimensions are already overt in the protocol provided.

Similarly, KSS0's induction tools would be appropriate if the relation between a person's attributes and a room's attributes were not known. The rules that the expert uses are available through empirical induction from enough cases. However, the knowledge/data tradeoffs involved are highly significant to the practicality of such induction (Gaines, 1989). The knowledge conveyed by the stereotypical cases provided by the expert is much greater than that from an arbitrary collection of data. Modeling the normative rules at work from stereotypical cases provided by an expert is simple and efficient. Doing the same thing from real-world examples, with anomalies, compromises, and so on, would require more data than is realistically available.

The current dataset provided an opportunity to test the most recent additions to the KSSn family described at JKAW'90 (Gaines, 1990a) and KAW'90 (Gaines, 1990b). KSSn is a designed as a set of modules communicating with KRS, a knowledge representation server with its KL-ONE-like representation capabilities. The modules include a graphic structure editor, KDraw, providing a visual language for knowledge entry and editing. The implementation and some applications of KRS have been described in detail elsewhere (Gaines, 1991a), as have its rule system (Gaines, 1991b), its visual language (Gaines, 1991c) and the underlying cognitive and logical principles and links to personal construct psychology (Gaines & Shaw, 1990). This paper focuses on experience in using these tools for the room allocation problem. Since this was the first major use of some of the new features, of particular interest were:

- How much of the problem could be encoded in a KL-ONE-like representation?
- What representation and inference mechanisms were used?
- What additional programming outside the knowledge base was required?
- How effective was the visual language as a knowledge entry and editing medium?
- How efficient was the visual language as an encoding medium compared with text?
- How useful was the type of solution in addressing some of the issues of under/over-determination discussed above, and in supporting the problem reformulation process?
- How general and generalizable was the solution developed?

Thus, the room allocation knowledge acquisition problem has been approached as one of the transcription of overt knowledge through an interactive graphic interface implementing a visual language for KL-ONE-like knowledge structures.

PROBLEM ANALYSIS

The expert's approach was seen to be one of classifying the people requiring rooms, classifying the rooms available, matching between the classes, proritizing the matches, making an allocation, and then re-classifying people and rooms to repeat the cycle. This is a *situated action* (Suchman, 1987) type of approach in which decisions are made as reactions to the current situation. The resultant actions so change the situation that the sequence of behavior can appear quite complex although it is conceptually based on the repetition of a single decision process without look-ahead or planning.

The following analysis is a direct transcript of notes made in analyzing the protocol.

Ground Rules

The protocol given is taken to be the primary source for acquiring a problem-solving strategy. It is assumed that this strategy should be generalized as much as possible to cope with other configurations of rooms and people.

The knowledge acquisition strategy is to use the visual language of KRS to:

1. Enter the knowledge structures in as terse and understandable form as possible
2. Express the problem-solving constraints in terms of these structures
3. Enter the facts of a particular situation
4. Make inferences in KRS that characterize the situation based on the constraints
5. Decide on the allocation of a person to a room based on these inferences
6. Enter this allocation as a fact and loop back to inference step 4

It seems that the problem can be treated as one of classification. If a situation satisfies a problem constraint then retrieve the rooms that satisfy a solution constraint.

Rules Relating to the Assignment of the Head of Group

Expert's Statement

The primary rule is that the Head of Group should be allocated a large central office.

Possible Generalizations

What should be done if there is no large central office?

Rules Relating to the Assignment of the Secretaries

Expert's Statement

The primary rule is that the secretaries should be close to the Head of Group.

A secondary rule is that secretaries should share offices if possible.

Possible Generalizations

What should be done if there is no office close to the Head of Group?

What should be done if there are several secretaries?

What should be done if one secretary is a smoker and another not? (cf researchers)

Rules Relating to the Assignment of the Manager

Expert's Statement

The primary rule is that the manager should have a small office close to the Head of Group and secretariat.

Possible Generalizations

What should be done if there is no small office close?

What should be done if the Head and secretariat are split?

What should be done if there is more than one manager?

Rules Relating to the Assignment of the Project Leaders

Expert's Statement

The primary rule is that the heads of large projects should have an office close to the Head of Group and secretariat, but the manager has higher priority.

These can be large rooms (however, presumably not if this would make the overall problem insoluble as in the given example).

Possible Generalizations

What should be done if there is no office close?

Rules Relating to the Assignment of the Research Staff

Expert's Statement

The primary rule is that if offices are shared smoking preferences should be the same.

A secondary rule is that if offices are shared the occupants should not be from the same project.

Possible Generalizations

Researchers may have specialist requirements, such as computer facilities, only available in particular offices.

Although the expert does not say it, the assignment of offices to minimize those from the same project being in the same office involves a more complex decision procedure, that those from the largest project should be assigned first, and so on.

Classification Structures

The control knowledge structures are seen to be lists of types of people without offices, and lists of offices suitable for such types of people, each created by heuristic classification.

The lists of people are:

- Head of Group without office
- Secretaries without office
- Manager without office
- Project Leaders without office
- Researchers without office
- Smoking researchers without office
- Non-smoking researchers without office

The lists of offices are:

- Large, central offices
- Large offices near Head of Group
- Large offices occupied by one secretary
- Single offices near Head of Group
- Large offices
- Large offices occupied by one smoking researcher
- Large offices occupied by one non-smoking researcher

The nearness of offices seems more subtle than an unchanging relation. What is wanted is something like "the nearest unoccupied offices to the Head of Group." This test is easily made.

Allocation Rules

The rules are given 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.

These eight rules seem to capture the problem.

SYSTEM ARCHITECTURE

Figure 1 shows the KSSn sub-systems that have been used to implement the solution developed. At the upper center is KRS (Gaines, 1991a), a knowledge representation server supporting KL-ONE-like capabilities. At the upper left is KDraw (Gaines, 1991c), a graphic structure editor for semantic networks providing a visual language for the entry of knowledge structures to KRS. At the upper right is a HyperCard stack controlling the problem solving and providing a user interface to the solution process. The three modules communicate through an inter-process communication protocol similar to that described by Gaines and Linster (1990) for KSS0-BABYLON integration.

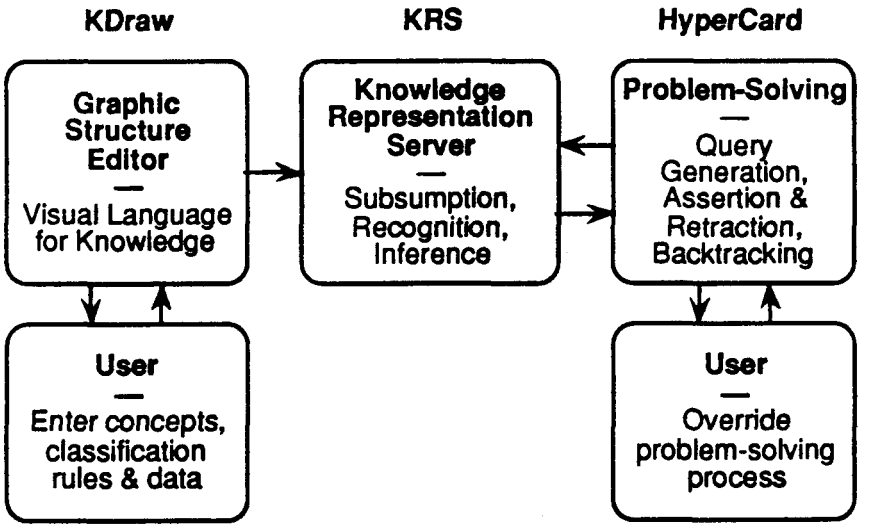

Fig.1 System architecture: KSSn sub-systems, functions and user interaction

The mode of operation is:

1. Domain concepts are entered through KDraw
2. Problem-solving classification rules are entered through KDraw
3. Specific problem facts are entered through KDraw
4. Knowledge base in visual language is compiled in KDraw and exported to KRS
5. User initiates problem-solving process from HyperCard

6. HyperCard queries KRS for room assignment recommendations
7. KRS runs inference and exports room assignment recommendations to HyperCard
8. User can accept default recommendation, select an equal priority one, or override and make any room assignment
9. HyperCard sends appropriate retractions and assertions to KRS and loops to 6
10. If a solution is not achieved, HyperCard recommends backtrack
11. User can accept backtrack or override it, selecting alternative backtrack, or making and retracting room assignments, and looping back to 6

The system can solve problems without user intervention, producing the first solution or all solutions, and backtracking as necessary. Note that the HyperCard function in the problem-solving is minimal. It acts to generate a sequence of selections among recommendations generated in KRS, and to generate alternative sequences by simple backtracking if required. This is a simple, highly generic problem-solving strategy that involves no domain knowledge. The "knowledge" that determines the effectiveness of this strategy is all within KRS, and is highly overt and easily edited through its visual representation in KDraw.

EXPRESSION IN THE VISUAL LANGUAGE

Figure 2 is an annotated screen dump of the concepts, data structure, agenda mechanism and rules for the solution of the room allocation problem. The graph of the knowledge structures has been shrunk by using a "fit to window size" display mode so that the details are not visible—the detailed knowledge structures are shown in the following figures. At the top of Figure 3 is shown the conceptual definitions splitting the domain into disjoint "knowledge islands" of people, organizations, locations and activities, and, at the bottom, the conceptual structure of an organization.

In the visual language, ellipses denote *concepts*, and those with small horizontal markers denote *primitive concepts* (those whose definitions are incomplete, such as *natural kinds*). A simple line between concepts defines them to be *disjoint*. An arrow from one concept to another defines the first concept to *inherit* the definition of the second (and hence to be subsumed by it). Text without a surround denotes a *role*. An arrow from a concept to a role makes that role part of the concept definition, together with any further constraints on the role—one chases arrow chains in compiling definitions. An arrow from a role to a concept defines fillers of that role in that particular definition to be constrained to be instances of the concept. Rounded corner boxes with short horizontal lines denote other *constraints* on role fillers, for example on cardinality or set membership.

Figure 4 shows the definition of types of employee and office in the visual language. A simple line between two roles indicates that they are *inverse* relations, e.g. that "occupant" and "occupies" are inverse to one another, and that "near" is symmetric. Figure 5 shows the assertions about the particular projects and employees involved in the instance of the problem given by the expert. A rectangle denotes an *individual*, and an arrow from it through a role to another individual asserts that the second individual fills the role in the first. An arrow from an individual to a concept asserts the individual to be constrained by the concept.

Figure 6 shows facts about individual employees being asserted. Since KRS makes no closed world assumptions it is necessary in the lower half to assert the initial state that each individual occupies no room and that the "occupies" role is closed (otherwise one is asserting that they may occupy a room but, if so, it is not known which one). Figure 7 shows facts about the rooms being asserted including the side, nearness to one another, and that all the rooms are initially unoccupied.

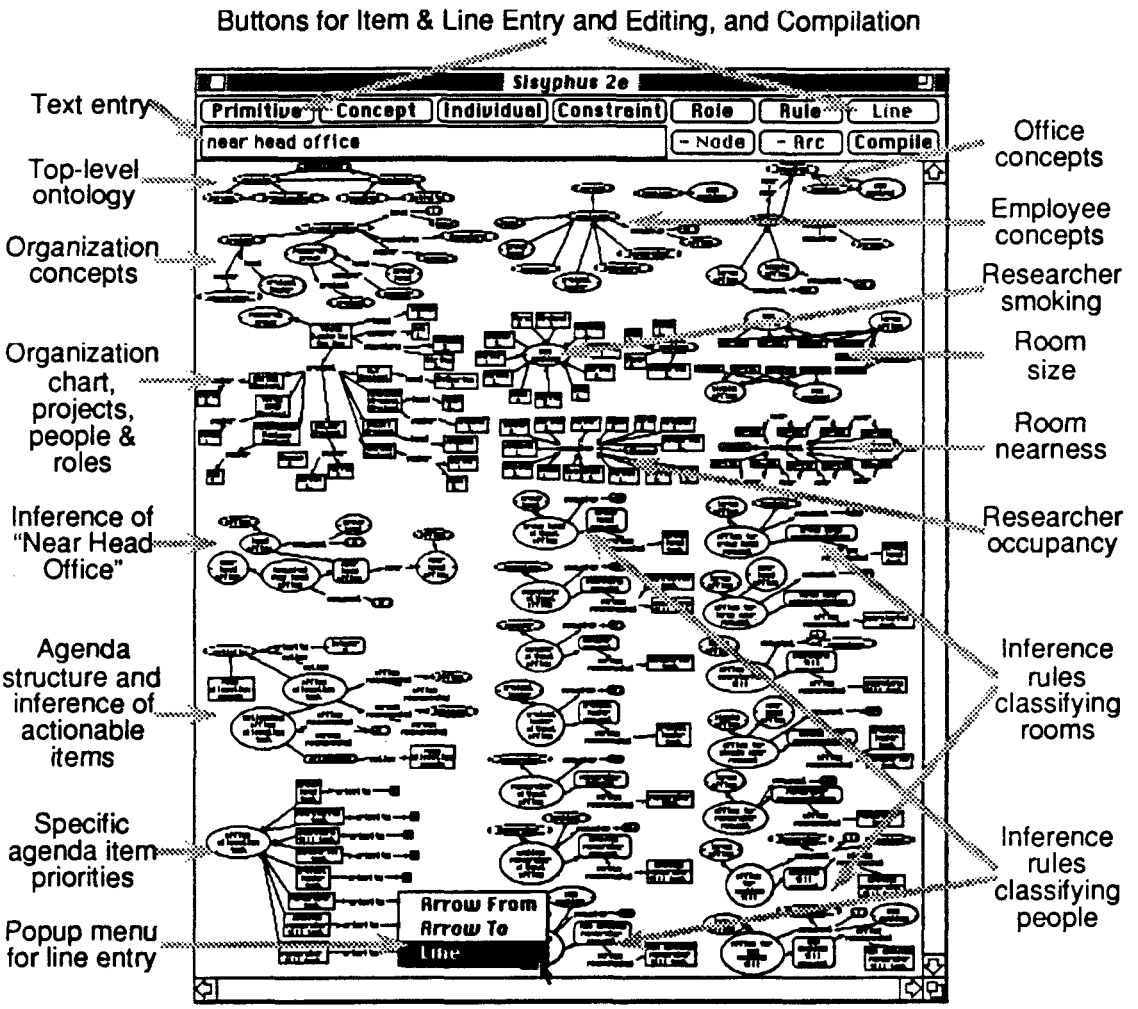

Fig.2 Problem, its solution and example dataset expressed in KDraw visual language

Figure 8 shows a pair of rules in the visual language. A rule is represented by a rounded corner box where an incoming arrow is from a concept definition triggering the rule, and outgoing arrows form a concept definition applying to the individual that triggers the rule (by being recognized as satisfying one of the concept definitions on the incoming arrows).

Thus, 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 is intended to capture the expert's apparent reclassification of offices further and further away from head office as being "near" to it as intervening offices are filled.

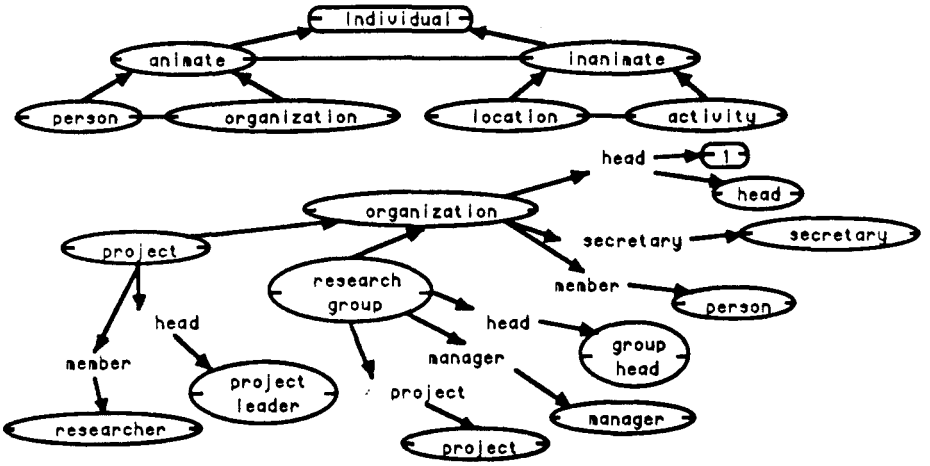

Fig.3 Top-level and organizational ontologies

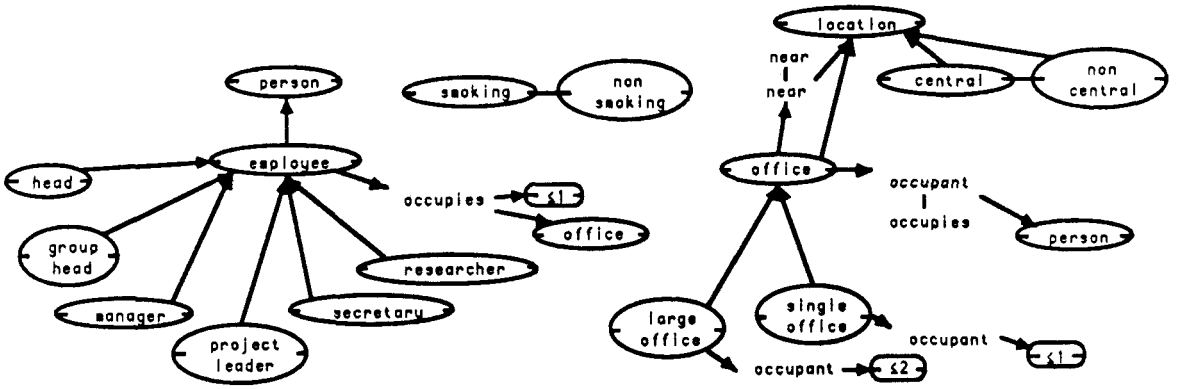

Fig.4 Employee and office ontologies

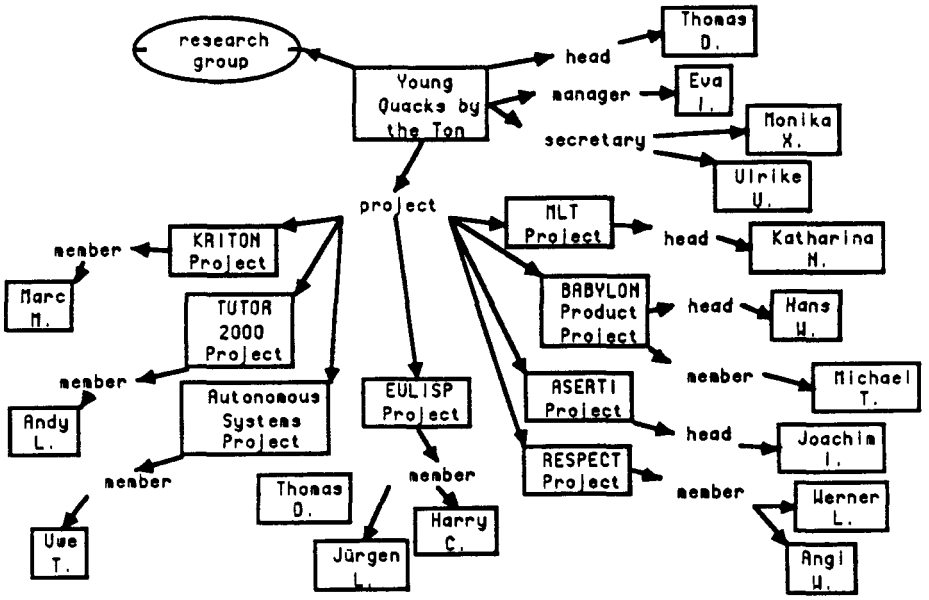

Fig.5 Facts asserted about projects and employees as an organization chart

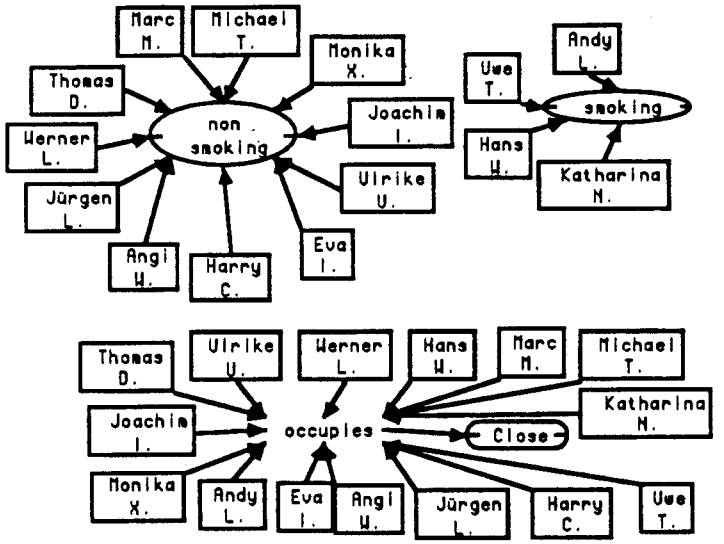

Fig.6 Facts asserted about employees

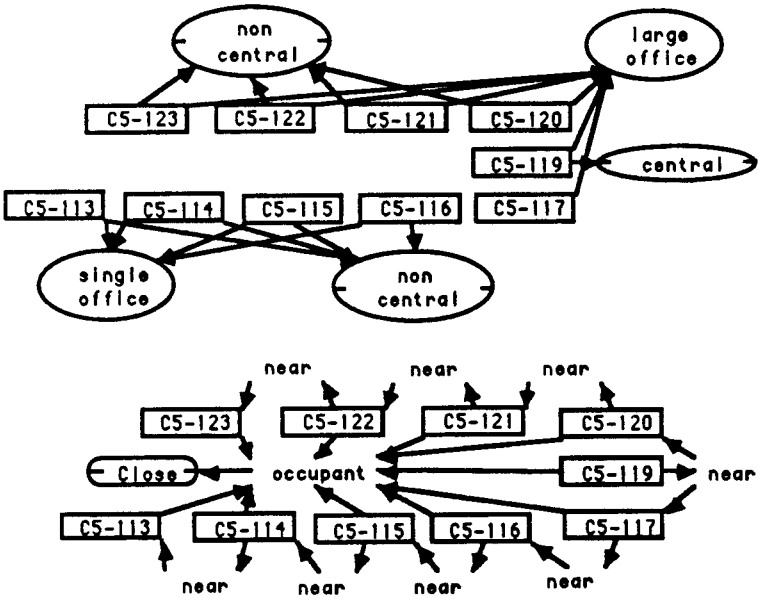

Fig.7 Facts asserted about rooms as a room layout plan

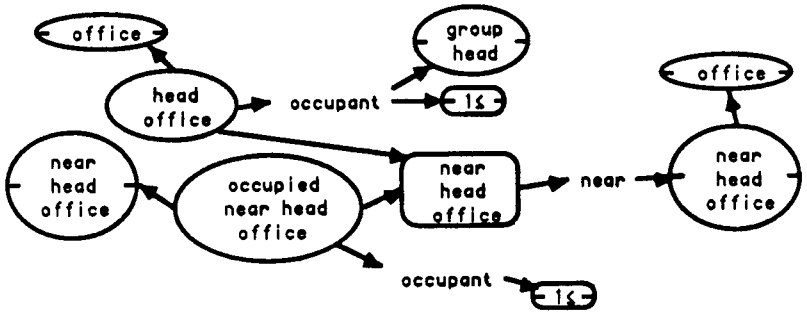

Fig.8 Rule inferring "near head office" and propagating it as offices fill

Figure 9 shows the agenda mechanism expressed in the visual language. Agenda items are "office allocation tasks" and have two roles, one to be filled by suitable employees and the other by suitable offices. The two roles are each defined as their own inverses since the mechanism for filling them will be through classification rules that place the agenda item in the equivalent role in an office or employee. Inference from the inverse relation then aggregates the offices and employees into the same roles in the agenda items. At the bottom of Figure 9 is a rule that places a possible agenda item on the actual agenda when it is "actionable", that is when there is at least one employee requiring the type of office and at least one office of the type available. Figure 10 defines the agenda items used in the expert's solution together with the priorities he seems to give to them.

Figure 11 gives the expert's problem solving strategy in terms of seven rules on the left that classify employees according to the types of room they need, and seven rules on the right that classify rooms in terms of the types of room that the classified employees need. The effect of using these rules to classify a particular configuration of employees and rooms at various stages of room allocation is to place a group of employees with a particular need in the "person recommended" slot of an agenda item, and to place groups of rooms satisfying that need in the "office recommended" slot of the same agenda item. The rule in Figure 9 then selects those agenda item that are actionable. No attempt is made within KRS to sort actionable agenda items by priority. This is left to an external mechanism using the classification in KRS to decide upon room assignments.

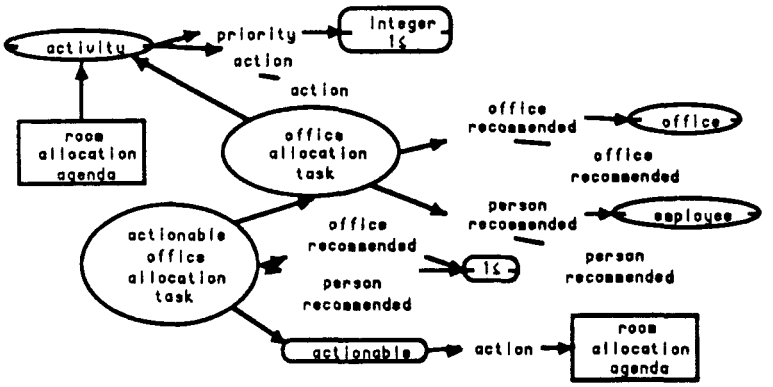

Fig.9 Concept definitions for agenda mechanism and rule putting items on it

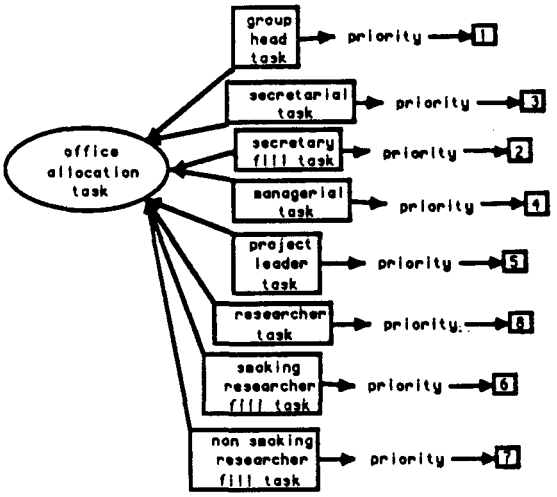

Fig.10 Priorities asserted for possible agenda items

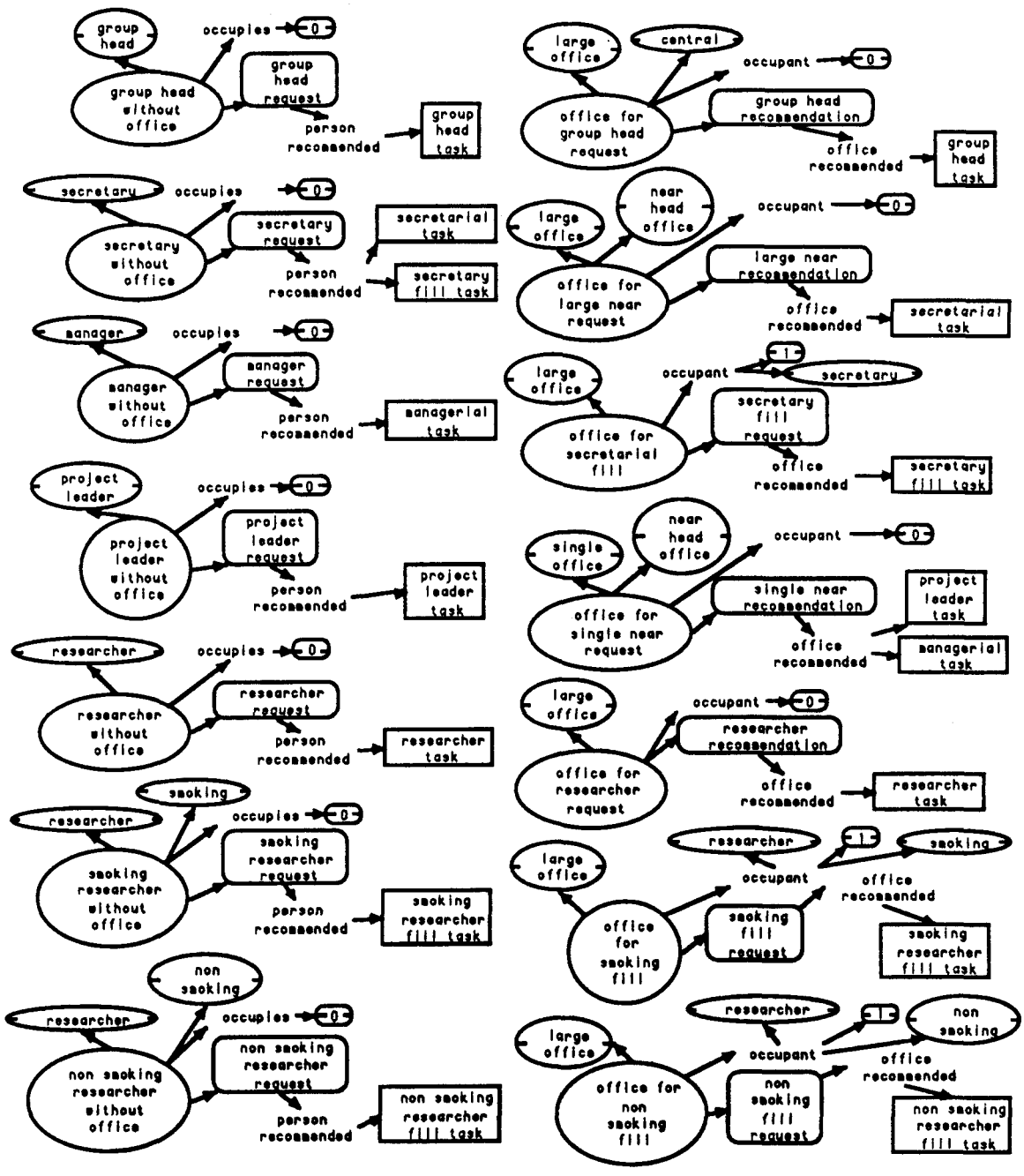

Fig.11 Rules classifying people (on left) and offices (on right)

The external decision making mechanism queries KRS for actionable agenda items, selects the highest priority one, and makes a room assignment for one of the employees to one of the rooms. If, as is usual, several possibilities are open, the mechanism can make an arbitrary choice, or refer to a human user. To allow for a sequence of assignments that reaches an impasse, the mechanism keeps track of past assignments and backtracks if necessary. The assignments and backtracks are reported to KRS as retractions and assertions of room occupants and rooms occupied. The KRS truth maintenance system handles the propagation of the consequences of both retractions and assertions.

A PROBLEM-SOLVING RUN

An example problem-solving run illustrates the operation of the system. The sequence of recommendations and assignments, given the dataset shown in Figures 2 through 11, is:

Recommendation 1
Highest priority agenda item: group head task
Person recommended: Thomas D.
Office recommended: C5-119
Assignment: Thomas D. to C5-119 (only possibility)

Recommendation 2
Highest priority agenda item: secretarial task
Person recommended: Monika X., Ulrike U.
Office recommended: C5-117, CS-120
Assignment: Monika X. to C5-117 (first pair out of four possibilities)

Recommendation 3
Highest priority agenda item: secretary fill task
Person recommended: Ulrike U.
Office recommended: C5-117
Assignment: Ulrike U. to C5-117 (only possibility)

Recommendation 4
Highest priority agenda item: managerial task
Person recommended: Eva I.
Office recommended: C5-116
Assignment: Eva I. to C5-116 (only possibility)

Recommendation 5
Highest priority agenda item: project leader task
Person recommended: Katharina N., Hans H., Joachim I.
Office recommended: C5-115
Assignment: Katharina N. to C5-115 (first pair out of three possibilities)

Recommendation 6
Highest priority agenda item: project leader task
Person recommended: Hans H., Joachim I.
Office recommended: C5-114
Assignment: Hans H. to C5-114 (first pair out of two possibilities)

Recommendation 7
Highest priority agenda item: project leader task
Person recommended: Joachim I.
Office recommended: C5-113
Assignment: Joachim I. to C5-113 (only possibility)

Recommendation 8
Highest priority agenda item: researcher task
Person recommended: Michael T., Werner L., Angi W., Harry C., Jürgen L., Uwe T., Andy L., Marc M.
Office recommended: C5-120, C5-121, C5-122, C5-123
Assignment: Michael T. to C5-120 (first pair out of thirty two possibilities)

Recommendation 9
Highest priority agenda item: non smoking researcher fill task
Person recommended: Werner L., Angi W., Harry C., Jürgen L., Marc M.
Office recommended: C5-120
Assignment: Werner L. to C5-120 (first pair out of five possibilities)

Recommendation 10
Highest priority agenda item: researcher task
Person recommended: Angi W., Harry C., Jürgen L., Uwe T., Andy L., Marc M.
Office recommended: C5-121, C5-122, C5-123
Assignment: Angi W. to C5-121 (first pair out of eighteen possibilities)

Recommendation 11
Highest priority agenda item: non smoking researcher fill task
Person recommended: Harry C., Jürgen L., Marc M.
Office recommended: C5-121
Assignment: Harry C. to C5-121 (first pair out of five possibilities)

Recommendation 12
Highest priority agenda item: researcher task
Person recommended: Jürgen L., Uwe T., Andy L., Marc M.
Office recommended: C5-122, C5-123
Assignment: Jürgen L. to C5-122 (first pair out of eight possibilities)

Recommendation 13
Highest priority agenda item: non smoking researcher fill task
Person recommended: Marc M.
Office recommended: C5-122
Assignment: Marc M. to C5-122 (only possibility)

Recommendation 14
Highest priority agenda item: researcher task
Person recommended: Uwe T., Andy L.
Office recommended: C5-123
Assignment: Uwe T. to C5-123 (first pair out of two possibilities)

Recommendation 15
Highest priority agenda item: smoking researcher fill task
Person recommended: Jürgen L.
Office recommended: C5-123
Assignment: Jürgen L. to C5-123 (only possibility)

This sequence solves the problem and no backtracking is required. The combinatorial explosion that will occur in enumerating all solutions that satisfy the constraints is most apparent in the assignments open at each stage. If a solution were impossible (for example, if Jürgen L. were non smoking) the sub-optimal possibilities open would also be apparent in the data generated.

COMMENTS

The system as described has trouble with Thomas D. being asserted to be both group head and a researcher in the EULISP project. It allocates him a large room as head and then later tries to put another researcher in it since the room satisfies the "office for non smoking fill" condition. It would have been possible to fix this by expanding the rules. However, it seemed more in the spirit of a visual interactive system when the problem came up to fix the facts by retracting Thomas D. being a member of EULISP—the arrow has been deleted in Figure 5. That is, the general question is do we want simple, understandable rules and human intervention in problem posing, or complex rules that take into account the combinatorial complexes possible in the world? There is clearly no universal answer, but it is an interesting and subtle design issue raised by this problem.

The requirement that researchers on the same project should not be placed in the same room has been omitted. It is a complex requirement to satisfy where an algorithmic approach seems more appropriate. It is a good example of where a "situated action" incremental approach breaks down. An effective algorithm first involves recognizing that researchers on the largest project should be allocated first. They will then be matched against researchers on the next largest project. However, although not stated by the expert, this would result in lack of diversity in project pairing. Hence the second step in a good strategy is to scatter the researchers on the second largest project among potential rooms. Overall, this part of the allocation problem seems best solved by an explicit algorithm in the external mechanism—it is not combinatorially complex.

There are related considerations in the global computation of whether the problem has no solution, and, if it has one, how much "slack" there is in allocation. Such computation indicates whether it is worth searching for a solution to the problem as stated, and whether, for example, rules allocating project leaders to large rooms should be activated. These "operations research" issues again seem best suited to an external mechanism.

GENERIC TASK OF RESOURCE ALLOCATION

The generic nature of the solution merits further comment. As noted initially, this study was not undertaken with a particular problem-solving approach in mind. However, the solution generalizes to other problems in which there are requirers of resources such that the requirers can be classified as wanting certain types of resource and the resources can be classified according to these types.

Figure 12 shows the generic task structure that may be abstracted. The knowledge structures that would be edited for variants on the problem are apparent. These range from the specific to the general—from the data for a particular case, through the rules of allocation to be applied, to the knowledge structures for the class of problems.

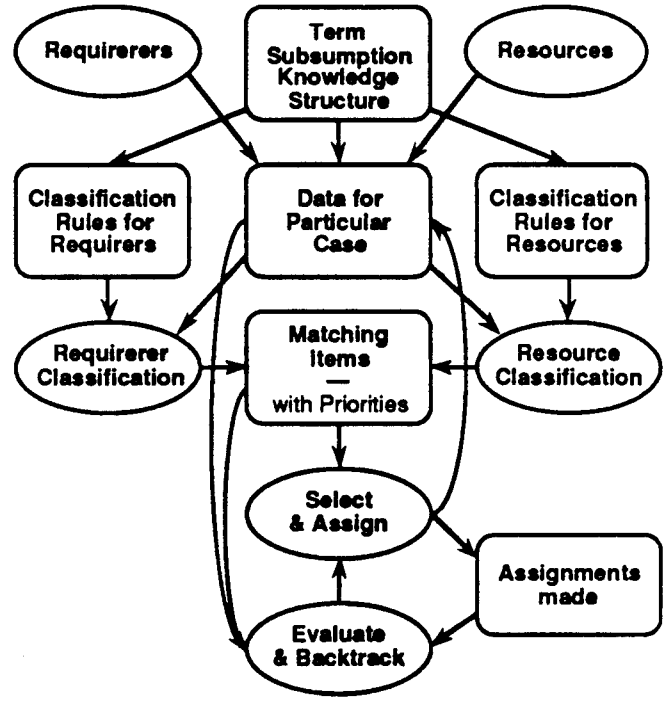

Fig.12 Generic task structure for resource allocation

CONCLUSIONS

The system described handles the class of problem defined in the Sisyphus documentation and generates room allocations that satisfy the constraints of the expert's example and variants of it. The solution 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

How peculiar these are to the approach taken in this paper is not clear. It will be interesting to test other approaches against these criteria, and to test this solution against attractive features of other approaches not considered here.

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

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

Bradshaw, J.M., Boose, J.H., Covington, S.P. & Russo, P.J., (1988) How to do with grids what people say you can't. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Third AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.3-1-3-15. Banff (November).

Ellman, T. (1989). Explanation-Based Learning. **ACM Computing Surveys, 21**(2) 163-221 (June).

Gaines, B.R. (1989) An ounce of knowledge is worth a ton of data: quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction. **Proceedings of the Sixth International Workshop on Machine Learning.** pp.156-159. Morgan Kaufmann.

Gaines, B.R. (1990a) Knowledge representation servers: a generic technology for knowledge acquisition and knowledge-based systems. **Proceedings of the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop: JKAW'90.** pp. 413-430. Tokyo: Ohmsha (October).

Gaines, B.R. (1990b) An architecture for integrated knowledge acquisition systems. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Fifth AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp. 8-1-8-22. Banff (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).

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

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. & 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).

Gaines, B.R. & Shaw, M.L.G. (1991). Eliciting knowledge and transferring it effectively to a knowledge-based systems. **IEEE Transactions on Knowledge and Data Engineering** to appear.

Suchman, L. (1987) **Plans and Situated Actions: The Problem of Human Machine Communication.** Cambridge: University 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).