A Situated Classification Solution of a Resource Allocation Task Represented in a Visual Language

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 solved using a situated classification approach. A solution was developed from the protocol provided in terms of three heuristic classification systems, one classifying people, another rooms, and another tasks on an agenda of recommended room allocations. The domain ontology, problem data, problem-solving method, and domain-specific classification rules, have each been represented in a visual language. These knowledge structures compile to statements in a term subsumption knowledge representation language, and are loaded and run in a knowledge representation server to solve the problem. The user interface has been designed to provide support for human intervention in under-determined and over-determined situations, allowing advantage to be taken of the additional choices available in the first case, and a compromise solution to be developed in the second.

1 INTRODUCTION

The Sisyphus room allocation problem is a resource allocation task in which a number of people with differing requirements as to type of room have to be allocated appropriate rooms from a number of rooms with differing characteristics. It was announced as a knowledge acquisition challenge problem that did not seem amenable to many of the existing knowledge acquisition techniques that focused on heuristic classification (Voß, Karbach, Drouven, Lorek and Schuckey, 1990). Such approaches appeared better suited to diagnostic tasks than to those involving serial allocation. Contrary to such analysis, the Knowledge Science Institute (KSI) representatives at the initial meeting proposed to develop a solution that showed the application of heuristic classification to this problem. One reason for accepting the challenge on these terms arose from the emphasis on classification in KSI tools such as KSS0 (Gaines and Shaw, 1993b) and KSSn (Gaines and Shaw, 1993a), and past claims that classification is fundamental to all knowledge processes. The sequential nature of the room allocation task, which superficially made it appear very different from classification, was seen to arise from the changing *situations* which arise as the task proceeds. That is, the classification process remains constant but the outcome changes with the changing situation, a *situated action* model of the task (Suchman, 1987; Clancey, 1993). In the solution presented in this paper, the sequence of actions necessary to solve the task originates from *situated classification* of the dynamic situation generated by actions derived from a static classification.

A second architectural decision was that the final solution should operate as completely as possible within a basic term subsumption classification system. Any meta-knowledge structures such as an agenda mechanism should be developed as part of the problem analysis and not just imported from an inference tool. The reason for this was two-fold: first, to ensure that the solution was completely overt and accessible for examination; and second, to allow the generic nature of the solution to be analyzed.

A third architectural decision was that careful attention should be paid to human involvement in the problem solving process. Resource allocation problems tend to be *over-determined*, and so notionally insoluble, or *under-determined*, and hence subject to combinatorial explosion. They move between the two extremes with minor changes to constraints or data, and usually have parts that are under-determined even if the whole is over-determined. 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. It was seen as important that the system developed be subject to manual override, and support the user in dealing with situations outside the scope of a logical solution. For example, that the options arising in under-determined situations should be made apparent to the user in such a way that advantage could be taken of the additional choices available, and that the conflicts resulting in the lack of a solution in the over-determined situation should be made apparent to the user in such a way that a compromise solution could be developed.

The problem was solved on these terms and the solution reported and demonstrated at EKAW'91. These results have been published in the context of enterprise modeling (Gaines, 1991d), and have also been used to illustrate the application of active document technology through parallel publication of a paper in which the knowledge base is embedded in paper and electronic forms (Gaines and Shaw, 1992). The electronic form opens in the KSSn word processing tool, KWrite, as a formatted document with full typography and embedded diagrams. Those diagrams which are knowledge structures in the visual language, KDraw, are active and can be edited within the document. The document can also be interrogated as a knowledge base in order to solve room allocation problems.

Given the availability of the original solution, this paper presents a later solution based on the same principles that copes with a wider range of problem variations.

2 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.

The knowledge structures in the dataset are already overt, and beyond the stage of knowledge modeling at which 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 such elicitation tools could be used to develop a knowledge structure: that is to indicate, given an exemplary set of people, or exemplary sets of rooms, how they differ on relevant dimensions. 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 room allocation dataset provided an opportunity to test the most recent additions to the KSSn family (Gaines, 1990). KSSn is a designed as a set of modules communicating with KRS, a knowledge representation server with KL-ONE-like (Brachman and Schmolze, 1985) term subsumption representation capabilities following those of CLASSIC (Borgida, Brachman, McGuiness and Resnick, 1989). The modules include a graphic structure editor, KDraw, providing a visual language for knowledge entry and editing that is more understandable for users than textual forms of KL-ONE-like knowledge structures (Nosek and Roth, 1990). The implementation and some applications of KRS have been described in detail elsewhere (Gaines, 1991a), as have its theoretical foundations (Gaines, 1993), rule system (Gaines, 1991b), its visual language (Gaines, 1991c) and the underlying cognitive and logical principles and links to personal construct psychology (Shaw and Gaines, 1992; Gaines and Shaw, 1993a).

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.

3 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, prioritizing the matches, making an allocation, and then re-classifying people and rooms to repeat the cycle. This is a situated action (Suchman, 1987; Clancey, 1993) 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.

3.1 Problem-Solving Strategy

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.

3.2 Rules Relating to the Assignment of the Head of Group

Expert’s Statement (Protocol 1)

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?

3.3 Rules Relating to the Assignment of the Secretaries

Expert’s Statement (Protocol 2)

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, not all of whom can be close?

What should be done if one secretary is a smoker and another not? (cf researchers, section 3.6)

3.4 Rules Relating to the Assignment of the Manager

Expert's Statement (Protocol 3)

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

Possible Generalizations

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

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

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

3.5 Rules Relating to the Assignment of the Project Leaders

Expert's Statement (Protocol 4, 5, 6)

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

It is also stated that a large room can be allocated but 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?

3.6 Rules Relating to the Assignment of the Research Staff

Expert's Statement (Protocol 7, 8, 9, 10)

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.

3.7 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:

P1. Head of Group without office (Protocol 1)

P2. Secretaries without office (Protocol 2)

P3. Manager without office (Protocol 3)

P4. Project Leaders without office (Protocol 4, 5, 6)

P5. Smoking researchers without office (Protocol 7)

P6. Non-smoking researchers without office (Protocol 8, 9, 10)

The lists of offices are:

O1. Large, central offices (Protocol 1)

O2. Large offices near Head of Group (Protocol 2)

O3. Large offices occupied by one secretary (Protocol 2)
O4. Single offices near Head of Group (Protocol 3, 4, 5, 6)
O5. Large offices (Protocol 7, 8, 9, 10)
O6. Large offices occupied by one smoking researcher (Protocol 7)
O7. Large offices occupied by one non-smoking researcher (Protocol 8, 9, 10)

The expert allocates pairs of individuals to large offices. In this implementation it was decided to do this in two stages so that the same rules could be used incrementally for situations that are not represented by the protocol. For example, where one person has left from a large office occupied by two people. This additional requirement motivates office classifications O3, O6 and O7 which make the solution more widely applicable, but do not lead to any deviation from the expert's decisions in the protocol case.

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 by recursive classification (see section 5.6).

3.8 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 (Protocol 1).

A large office already occupied by a secretary should be allocated to another secretary (Protocol 2).

A large office near the Head of Group is suitable for the secretaries (Protocol 2).

A single office near the Head of Group is suitable for the Manager (Protocol 3).

A single office near the Head of Group is suitable for the Project Leaders (Protocol 4, 5, 6).

A large office with a smoking researcher occupant should be allocated to another smoking researcher (Protocol 7).

A large office should be allocated to smoking researchers (Protocol 7).

A large office with a non-smoking researcher occupant should be allocated to another non-smoking researcher (Protocol 8, 9, 10).

A large office should be allocated to non-smoking researchers (Protocol 8, 9, 10).

These eight rules seem to capture the essence of the problem.

The requirement that two researchers on the same project, if possible, not share a room requires different treatment. It is interaction between pairs of researchers dependent on a specific value, not a general concept such as smoking. It is a desirable rather than essential constraint. It applies when one researcher is being allocated to a room already occupied by another and is best handled by removing, or marking as undesirable, those researchers who qualify for the room but are members of the same project as the existing occupant. This involves a third classification, that of the agenda items generated from the primary rules as described in section 5.5.

4 SYSTEM ARCHITECTURE

Figure 1 shows the KSSn sub-systems that have been used to implement the solution developed. At the upper center is KRS, the knowledge representation server providing KL-ONE-like capabilities. At the upper left is KDraw, the 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.

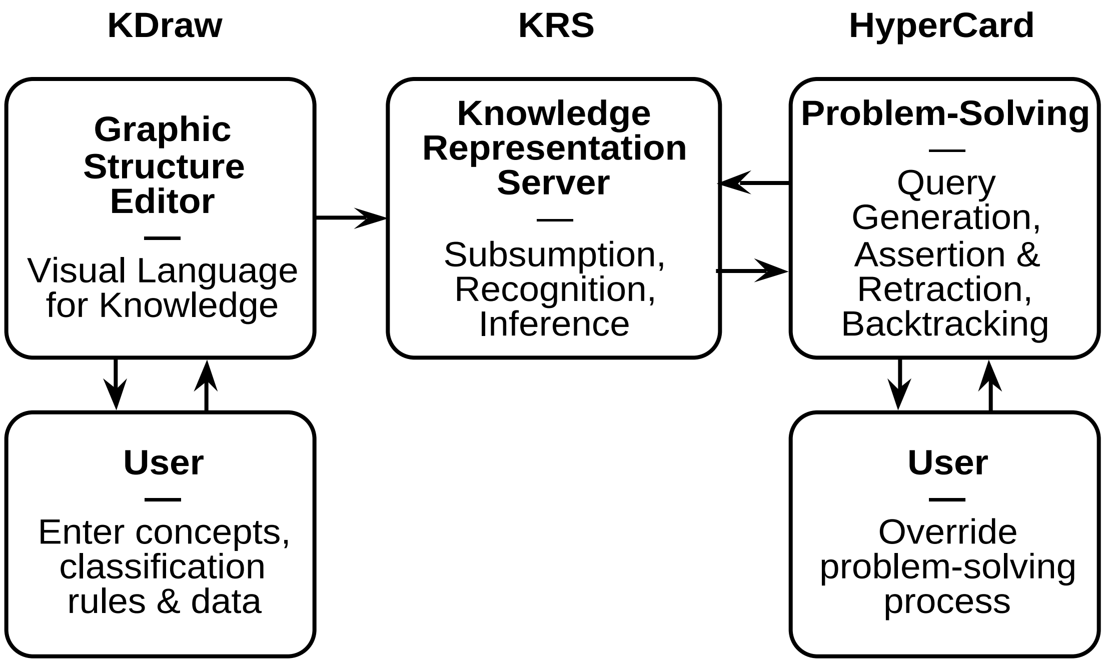

KDraw
KRS
HyperCard
Graphic Structure Editor — Visual Language for Knowledge
Knowledge Representation Server — Subsumption, Recognition, Inference
Problem-Solving — Query Generation, Assertion & Retraction, Backtracking
User — Enter concepts, classification rules & data
User — Override problem-solving process

Figure 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.

5 KNOWLEDGE STRUCTURES

This section describes the detailed knowledge structures used to implement the solution developed in section 3, and shows them as entered in the visual language.

5.1 The Visual Language

The concepts, rules and data necessary to operationalize the problem analysis are entered directly in the visual language using KDraw. Figures 3 through 6 show the top level problem ontology, employee ontology, office ontology, and organization ontology, respectively.

The visual language used in these figures is precisely defined. *Concepts* are ovals, *primitive* concepts are ovals with small horizontal lines inside each side, *individuals* rectangles, *roles* unboxed text, *constraint* expressions rounded-corner boxes, and *rules* (in later figures) rectangles with double lines at each end. 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

5.2 Editing the Visual Language

Figure 2 shows the definition of types of organization being edited in KDraw, the graphic editor for the visual language. 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. Nodes are entered by typing their content into the text box at the top and clicking the appropriate button. The syntax of possible node interconnections and constraint expressions is enforced—it is not possible to enter a graph that is syntactically incorrect.

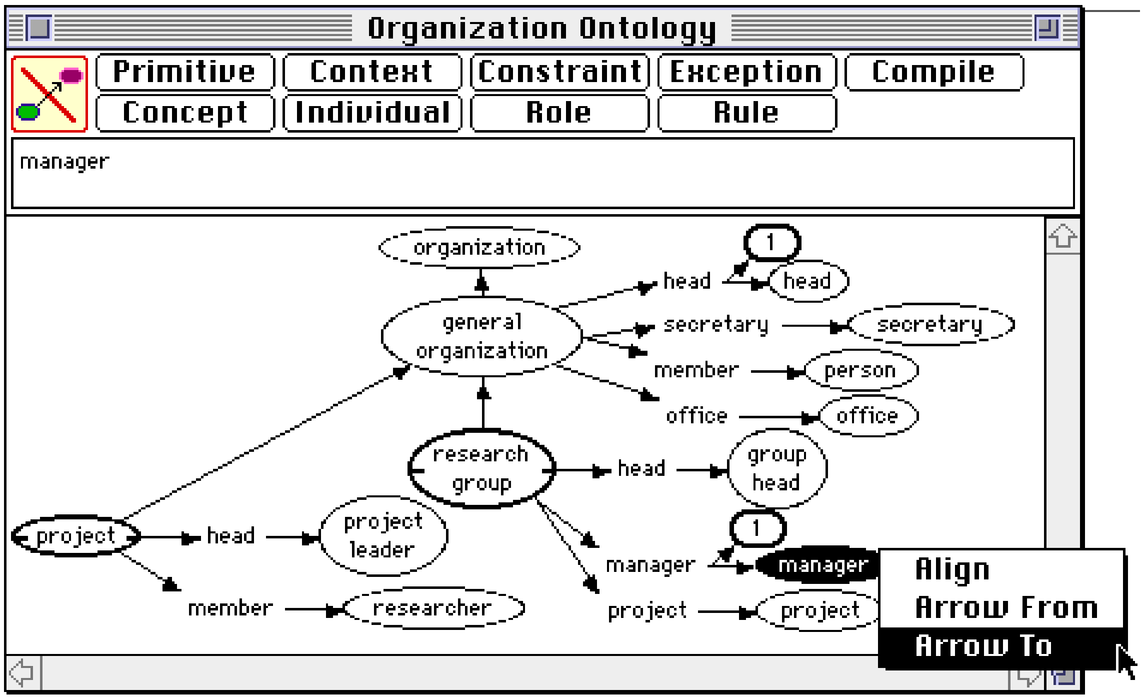


Figure 2 Editing the organization ontology in KDraw

When the cursor is at the edge of a node it changes to a connection arrow, and an arrow can then be drawn from that node to another by mousing down and dragging until over the other node. Alternatively the pop-up menu shown at the lower right may be used to connect and align nodes. Double-clicking on an existing node opens it for editing. Cut-and-paste of graphs and subgraphs is supported. 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.

5.3 Definition of the Domain Ontology

The top level problem ontology in Figure 3 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 constraint “Individual” at the top is a type constraint that constrains the concept to apply only to domain individuals, not, for example, to integers.

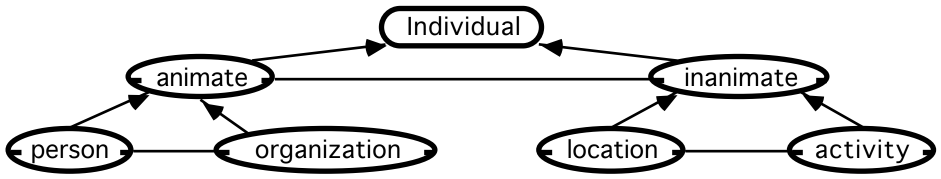


Figure 3 Top level problem ontology

The ontology essentially divides the domain into four disjoint kinds of individual, people, organizations, locations and activities. The only function of this division is to speed classification in that, for example, concepts that apply only to people will quickly be found not to apply to location such as office. There is often deep debate about such top-level ontologies and they can appear to raise deep linguistic or philosophical problems. However, pragmatically, their role is to represent the disjoint sub-domains of the problem domain, and how this is done, and the terminology used, are otherwise irrelevant.

Figure 4 shows the definition of types of employee in the visual language. “Employee” inherits from “person” and hence is automatically inferred to be disjoint from “organization”, “location” and “activity.” It gains the further attribute of having an “occupies” role filled by at most one individual of type “office”, i.e. an employee can only occupy at most one office. The types of employee, “head”, “group head”, and so on, inherit these properties of employee and are further defined to be primitive in their own right. We cannot recognize a “group head” individual from its properties. It has to be specifically asserted to be one.

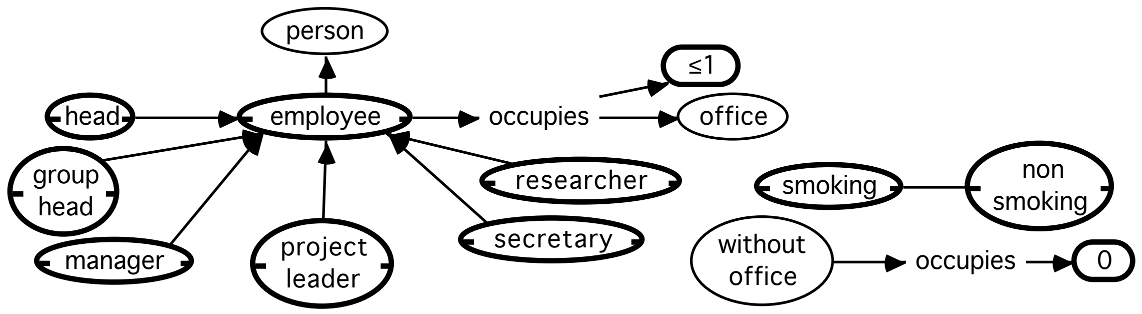


Figure 4 Employee ontology

The bipolar construct “smoking—non-smoking” is defined through a pair of disjoint primitive concepts. It is not made a sub-concept of person since it is useful to be able to apply it to rooms also. The concept “without office” is defined specifically because it is used in a number of rules. It is defined by a cardinality constraint, as having the “occupies” role filled with zero individuals.

Figure 5 shows the definition of types of office in the visual language. An “office” is a “location” and hence disjoint from “person”, “organization” and “activity.” It can be “near” another location, and it may have an “occupant” who is a “person.” 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. A “single office” is constrained to have at most one occupant, and a “large office” at most two. The concept “unoccupied” is defined by a cardinality constraint, as having the “occupant” role filled with zero individuals. Locations divide into two disjoint sub-concepts, “central” and “non-central”.

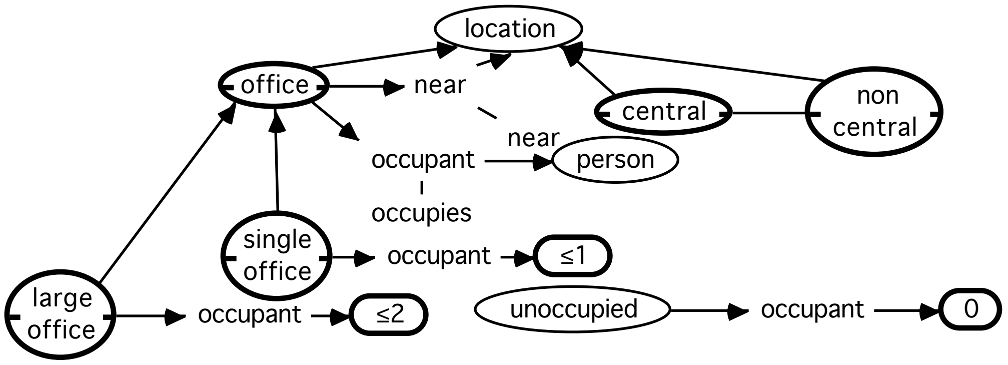


Figure 5 Office ontology

Figure 6 shows the definition of types of organization in the visual language. A “general organization” is an “organization” and hence disjoint from “person”, “location” and “activity”. It has exactly one “head” who is a “head”, an indefinite number of fillers of the “secretary” role who are of type “secretary”, an indefinite number of fillers of the “office” role who are of type “office”, and an indefinite number of fillers of the “member” role who are of type “person”.

A “project” is a “general organization” whose “head” is a “project leader” and whose “member” role is filled by those of type “researcher”. A “research group” is a “general organization” whose “head” is a “group head”, which has exactly one “manager” of type “manager”, and whose “member” role is filled by an indefinite number of individuals of type “project”.

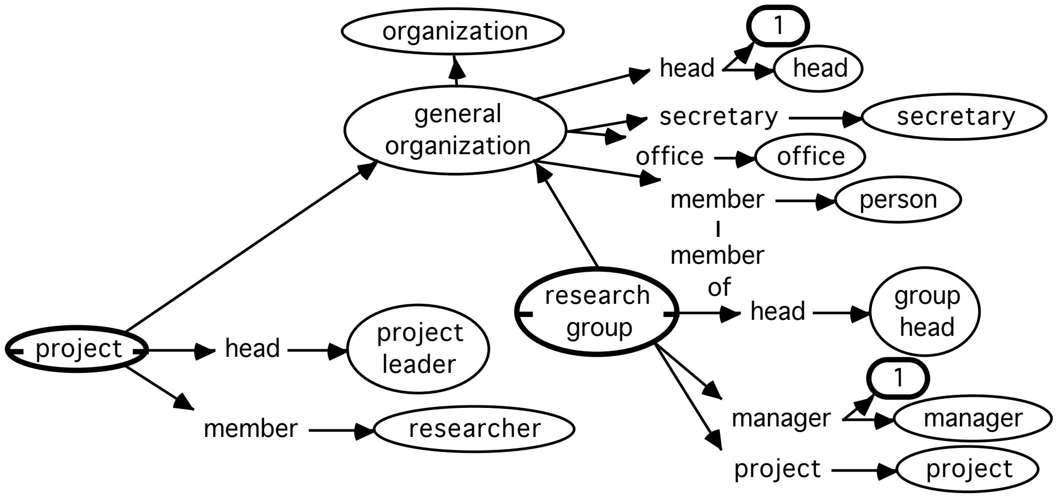


Figure 6 Organization ontology

5.4 *Assertions about Problem Data*

Figure 7 shows the assertions about the particular projects and employees involved in the instance of the problem given to 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. The "close" constraint on the secretary role means that Monika and Ulrike are the only secretaries. This is necessary if rules test whether all secretaries have been allocated offices since the inference process in KRS makes no closed world assumptions.

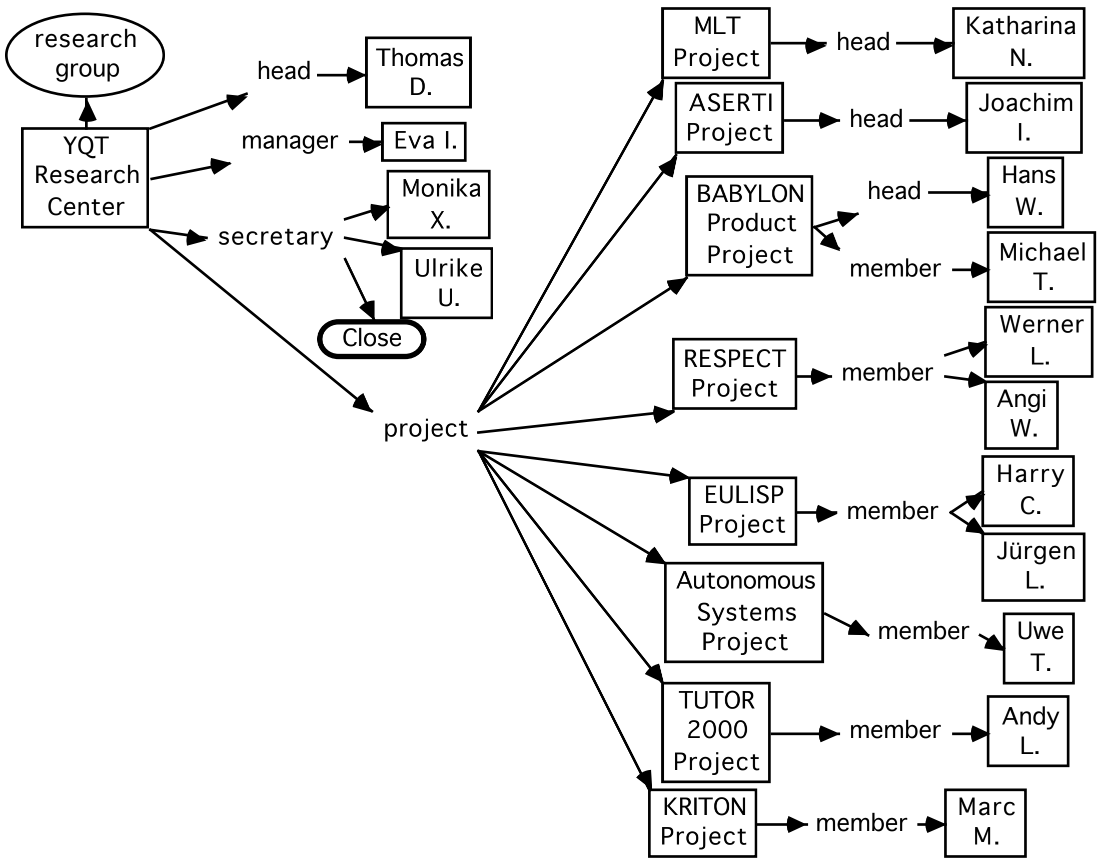


Figure 7 Organization chart

The visual language allows the actual organization to be represented in a way that resembles a normal organization chart and, hence, should be easily understood and edited. Note that only one concept appears, "research group" at the upper left. The assertion that "YQT Research Center" is an instance of "research group" is sufficient for the constraints in the knowledge structures in Figures 3 through 6 to be applied. For example, "Thomas D." will be inferred to be "animate", "person", "employee", "head" and "group head". Similarly, "Eva I." will be inferred to be "animate", "person", "employee" and "manager". The "MLT Project" will be inferred to be "animate", "organization" and "project". "Katharina N." will be inferred to be "animate", "person", "employee", "head" and "project leader".

Figure 8 shows additional assertions about employees specifying whether they smoke or not. In practice this information might come from a database of employee records.

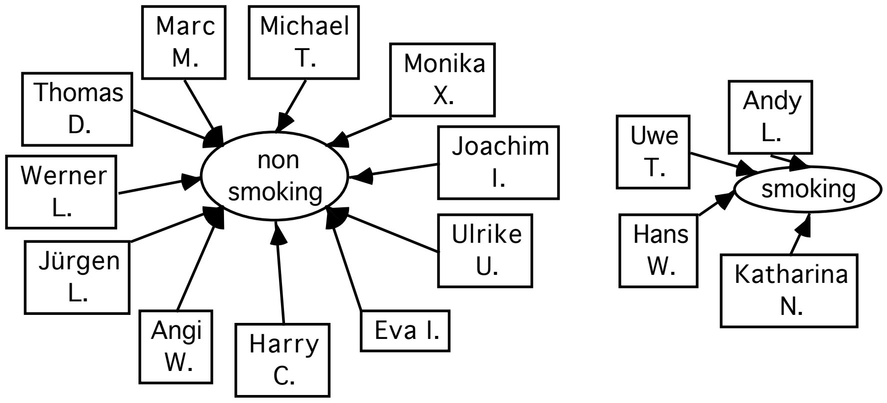


Figure 8 Employees smoking

Figure 9 shows the assertions about the particular rooms involved in the instance of the problem given to the expert. The visual language allows the actual organization to be represented in away that resembles a normal floor plan and, hence, should be easily understood and edited. In practice the information about centrality, size and research group allocation might come from a database of accommodation records.

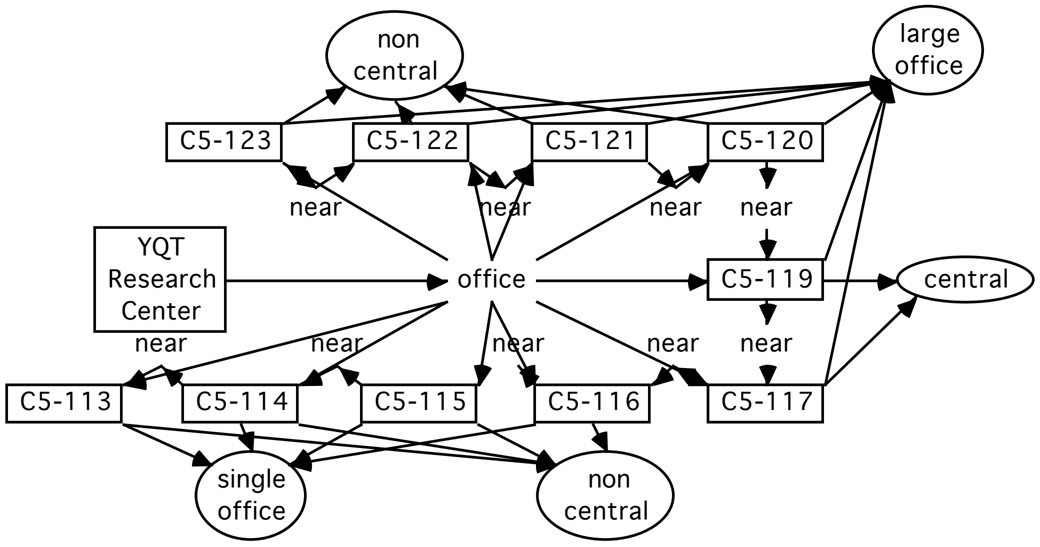


Figure 9 Room layout

Figure 10 asserts the initial conditions that no-one occupies an office and that no offices have occupants.

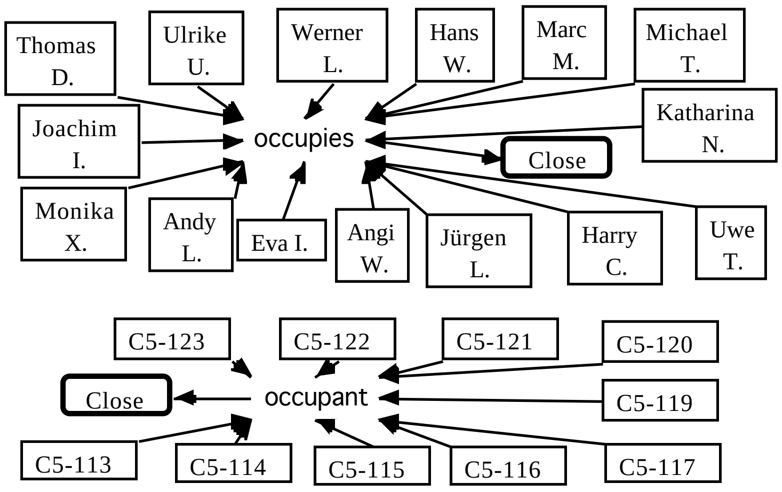


Figure 10 Initial facts

5.5 The Agenda-Based Problem-Solving Method

Figures 11 through 13 show the knowledge structures for the agenda-based problem-solving method. In Figure 11, a “task” is defined as an “activity” with a priority specified as an integer greater than or equal to 1. An “office allocation task” is defined to have a self-inverse “office recommended” role filled by an “office”, and a self-inverse “person recommended” role filled by an “employee”. Four primitive sub-concepts of “office allocation task” are defined, “priority allocation task”, “available allocation task”, “suitability allocation task”, and “pair allocation task”.

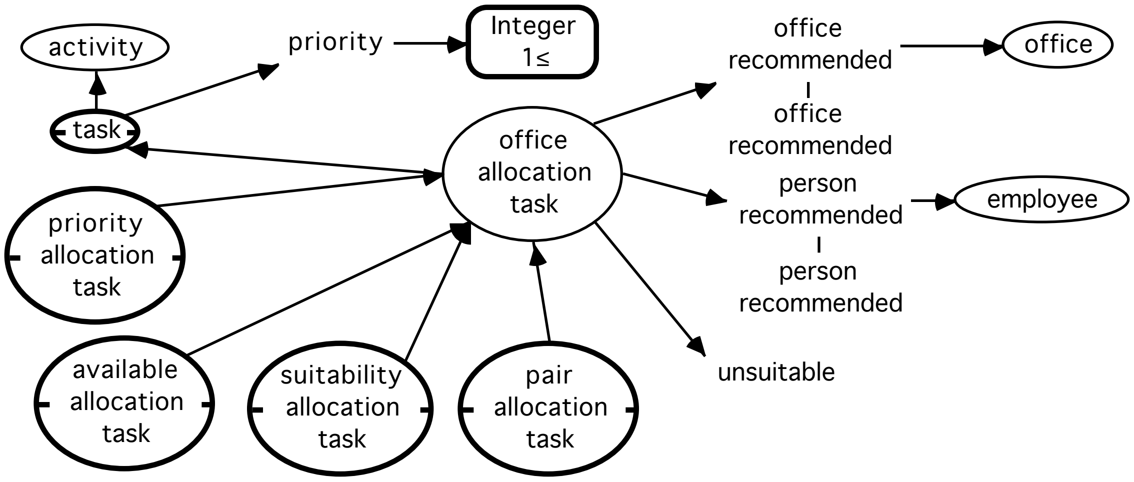


Figure 11 Agenda ontology

In Figure 12, ten tasks are defined as instances of these sub-concepts, and priorities assigned to them. Nine of these tasks correspond to the allocation rules given in Section 3.9, and the tenth, the “all without task”, to an agenda item corresponding to all those without rooms which will be used to allow the allocation recommendations to be overridden by the user if required.

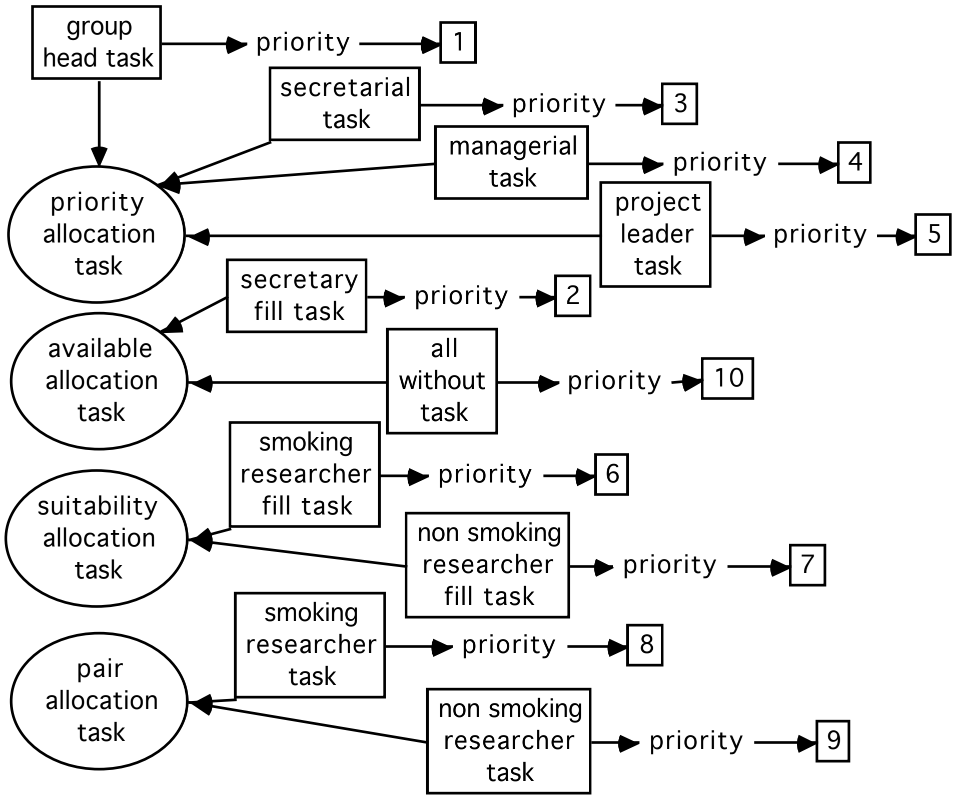


Figure 12 Agenda priorities

In Figure 13, the rules that an agenda item is “actionable” or that some rooms are “unsuitable” are specified. If an agenda item is “actionable” it is classified as being “on agenda”. There are four concepts specifying when an is “actionable”. First, for a “priority allocation task” with a person recommended. This corresponds to the need to allocate a room to a group head, secretary, manager or project leader, even if no suitable room is available. Second, for an “available allocation task” with both a person and an office recommended. This corresponds to a normal allocation task on the agenda. Third, for a “pair allocation task” with at least two people and an office recommended. This corresponds to the pairwise allocation of researchers to rooms. Fourth, for a “suitability allocation task” with both a person and an office recommended. This corresponds to the need to fill a room with a second researcher and to check the suitability in that someone from the same project should not be allocated. This latter is done by the co-reference clause on the right-hand side of the “suitability” rules that specifies that the “unsuitable” role contains the contents of the “member” role of the those filling the “member of” role of the “office recommended” role of the agenda item. The effect is that the “unsuitable” role of the agenda item is filled with the employees who are members of the same project as the person already occupying the room.

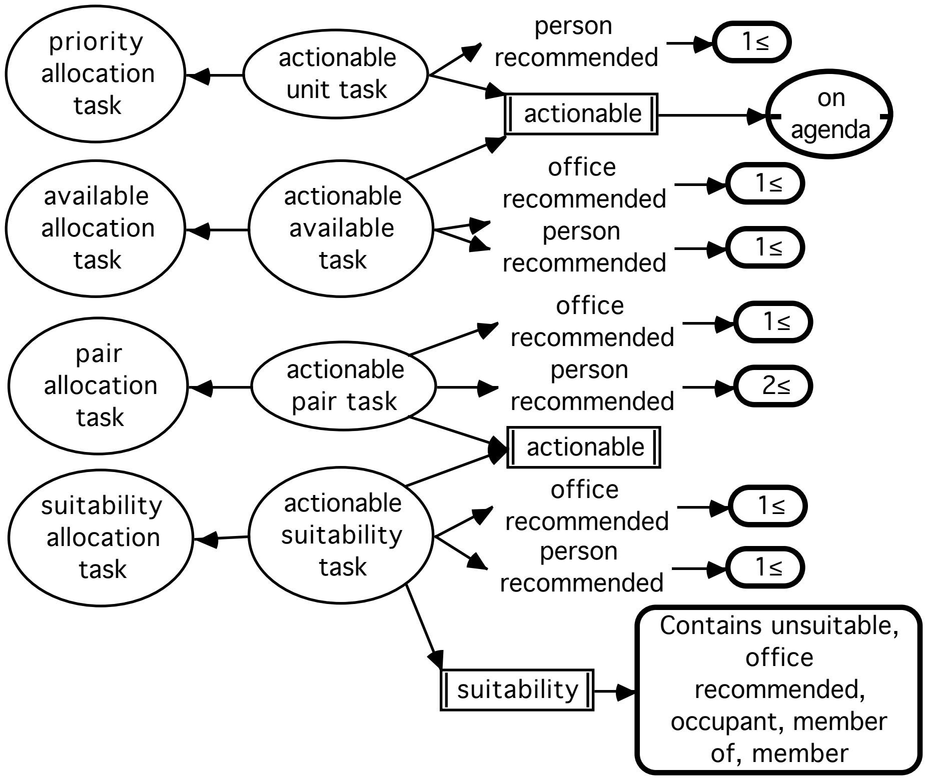


Figure 13 Agenda rules

5.6 The Domain Classification Rules

Figure 14 shows three rules which classify an office as being near head office. The top concept, “head office”, is defined as an office that has one occupant who is a “group head”. The rule, “near head office”, then asserts that an office near “head office” starts a “large chain” and a “single chain”. The two lower rules are called recursively to propagate the chains past occupied offices. The result is to find the first unoccupied single offices and large offices nearest head office.

Figure 15 shows seven rules classifying employees corresponding to the employee classification rules in Section 3.7. The effect of these rules is to place the appropriate agenda item in the “person recommended” role of the person, and hence, since this is a self-inverse role, this also place the person in the “person recommended” role of the agenda item.

Figure 16 shows eight rules classifying offices corresponding to the office classification rules in Section 3.7. The effect of these rules is to place the appropriate agenda item in the “office recommended” role of the office, and hence, since this is a self-inverse role, to also place the office in the “office recommended” role of the agenda item.

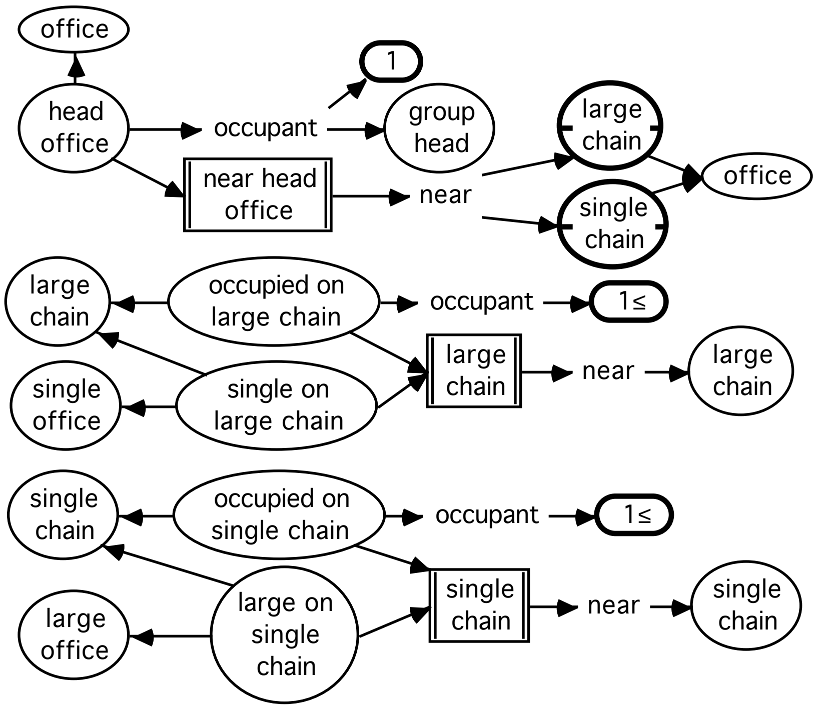


Figure 14 Near head office rules

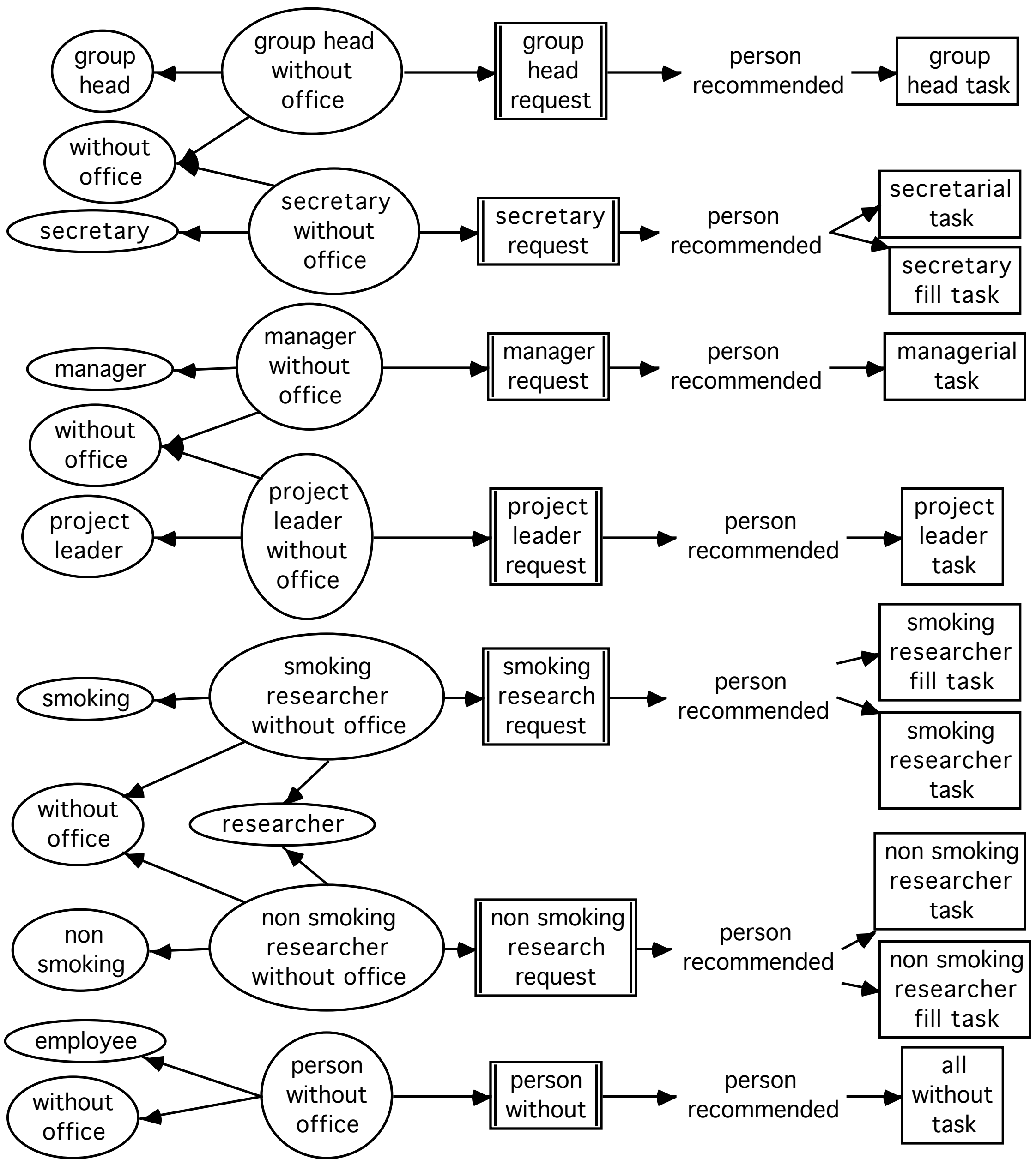


Figure 15 Employee rules

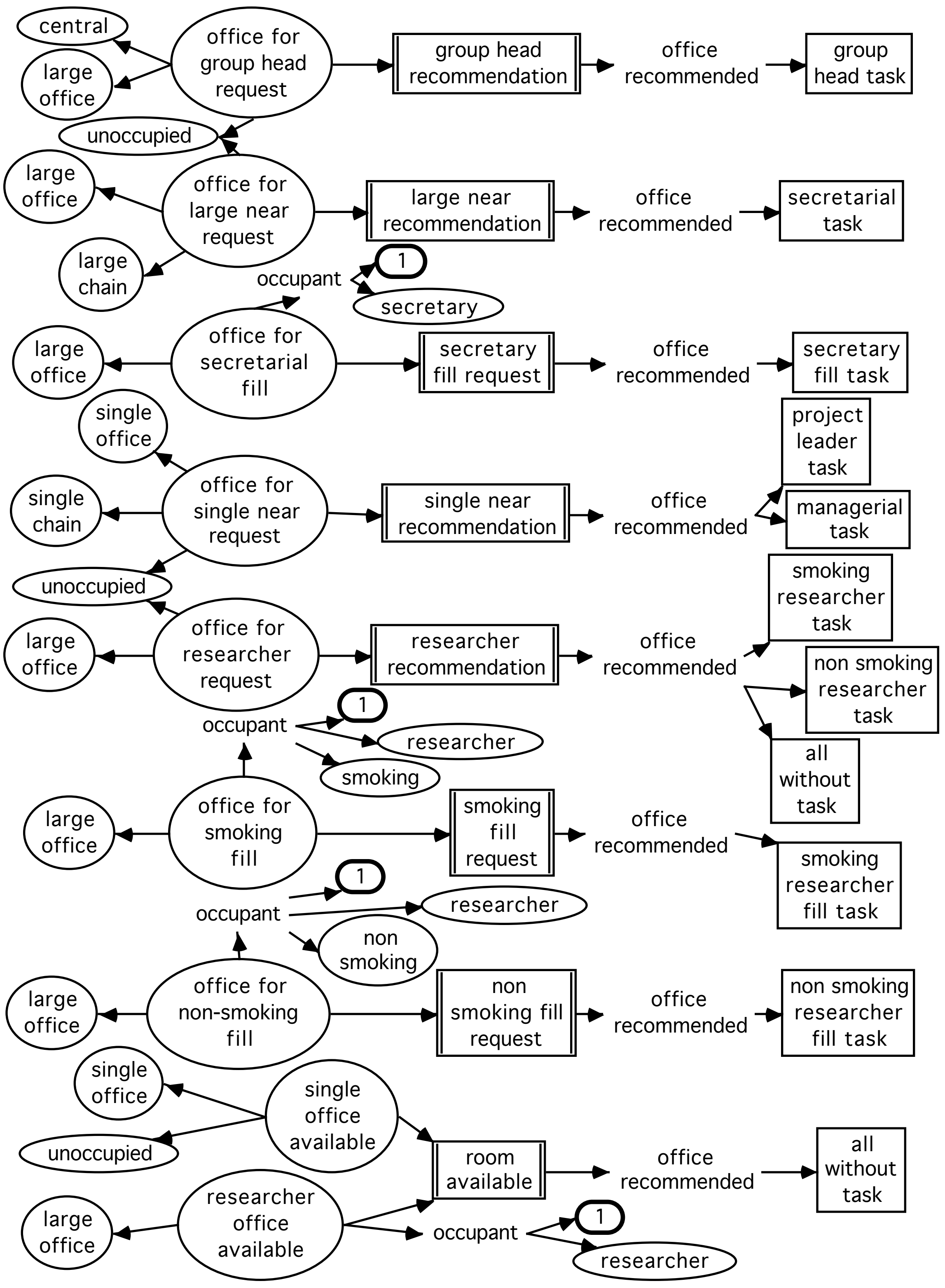


Figure 16 Office rules

5.7 Summary and Operation

Thus, overall, the knowledge structures in Figures 3 through 6 define the domain ontology, and those in Figures 7 through 9 assert the facts of a particular domain. Those in Figures 11 through 13 define concepts and assert facts that determine the problem solving methodology, and those in Figures 14 through 16 determine how this methodology will be applied to the domain.

KRS automatically links knowledge structures so that concepts defined in one module may be used in another, and hence the knowledge structures in the figures may be loaded independently in the order presented.

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 can backtrack if necessary. The backtracks and new assignments 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.

6 THE USER'S PROBLEM-SOLVING INTERFACE

As shown on the right of Figure 1, the end user's problem solving interactions with the knowledge structures is interfaced through HyperCard which uses KRS as a server. 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 17 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.

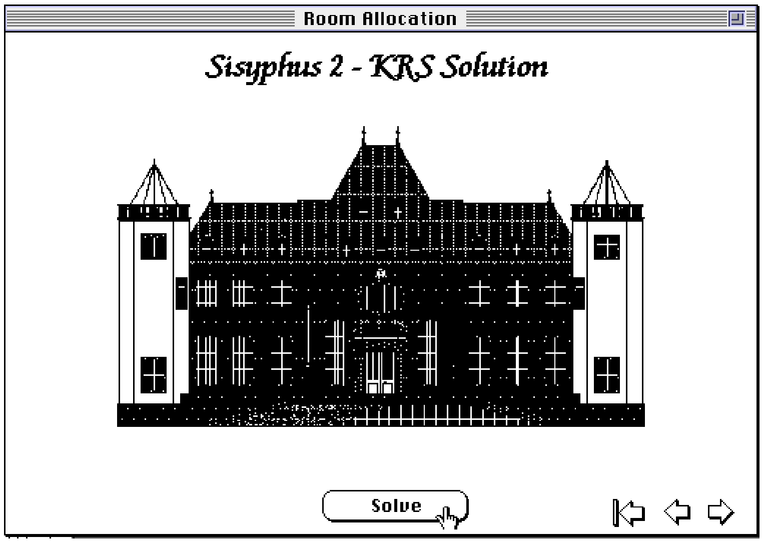
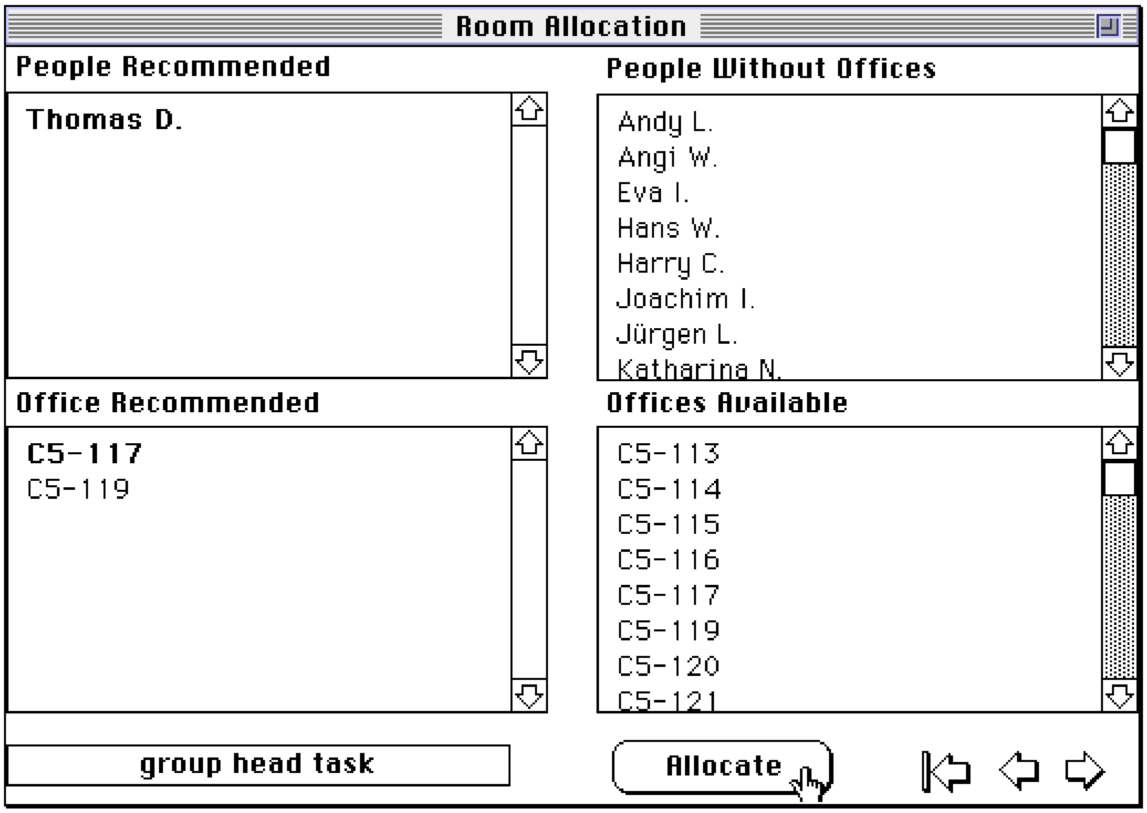


Figure 17 Initial HyperCard screen

Figure 18 shows the screen through which allocation occurs with the first suggestion appearing of an allocation of the group head. The top left field shows the people recommended, in this case just “Thomas D.” The field below it shows the rooms recommended, in this case “CS-117” or “CS-119”. The field below it shows the task generating these recommendations, in this case the “group head task”. The top field on the right shows all the people without offices, and the field below it shows all the offices available. Clicking on the name of a person in one of the top two fields, or the name of a room in one of the two fields below, causes the name to be highlighted in bold.

If both a person and an office are selected the “Allocate” button at the lower center right appears, and clicking on this causes this allocation to be sent to the KRS server and the inference to proceed.

By default, the first person in the people recommended field and the first office in the office recommended field are selected, but all possible combinations of these two fields satisfy the rules and are equally valid. The user may chose between them using other criteria not included in the knowledge structures. The user may also override the rules and chose a combination from the right hand fields of any person without an office and any available office.



Thomas D.

Andy L.
Angi W.
Eva I.
Hans W.
Harry C.
Joachim I.
Jürgen L.
Katharina N.

CS-117
CS-119

CS-113
CS-114
CS-115
CS-116
CS-117
CS-119
CS-120
CS-121

group head task

Figure 18 First allocation suggestion

Figure 19 shows the message sent from HyperCard to KRS when the “Allocate” button is clicked. The first character is a delimiter, and the first line is a command to KSSn to send the body of the message to the KRS server instance “Room Allocation”, returning the message “Solve” as a header to the reply. The body of the message is a series of commands to KRS in relation to the “Room Allocation” knowledge structures. It is first retracted that “CS-117” has no occupant and “Thomas D.” occupies no room, and then the new allocation is asserted. The four statements are excessive in that either one of the last two will have the effect of all four since “occupies” and “occupant” are inverse, and the need for retraction is inferred automatically. However, the full set of statements is clearer in demonstrations. Then inference is run and a query made for individuals classified as “on agenda”.

```
|$KRS|Room Allocation|Solve|
Individual(C5-117, (Retract occupant))
Individual(Thomas D., (Retract occupies))
Individual(C5-117, (Close occupant, Thomas D.))
Individual(Thomas D., (Close occupies,C5-117))
Infer()
Query(individuals(on agenda))
```

Figure 19 Message from HyperCard to Knowledge Representation Server

Figure 20 shows a recommendation at a later stage when a non smoking researcher is needed to fill a room already occupied by “Angi W.”. The “suitability” rule at the bottom of Figure 13 has operated to exclude “Werner L.” from the list of people recommended because is on the same project as “Angi W.”

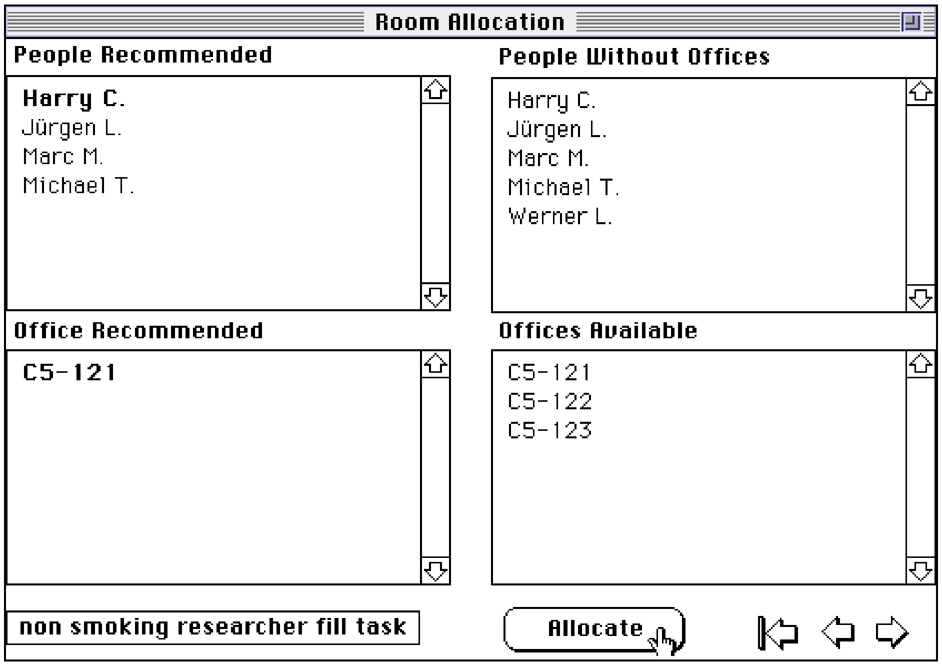
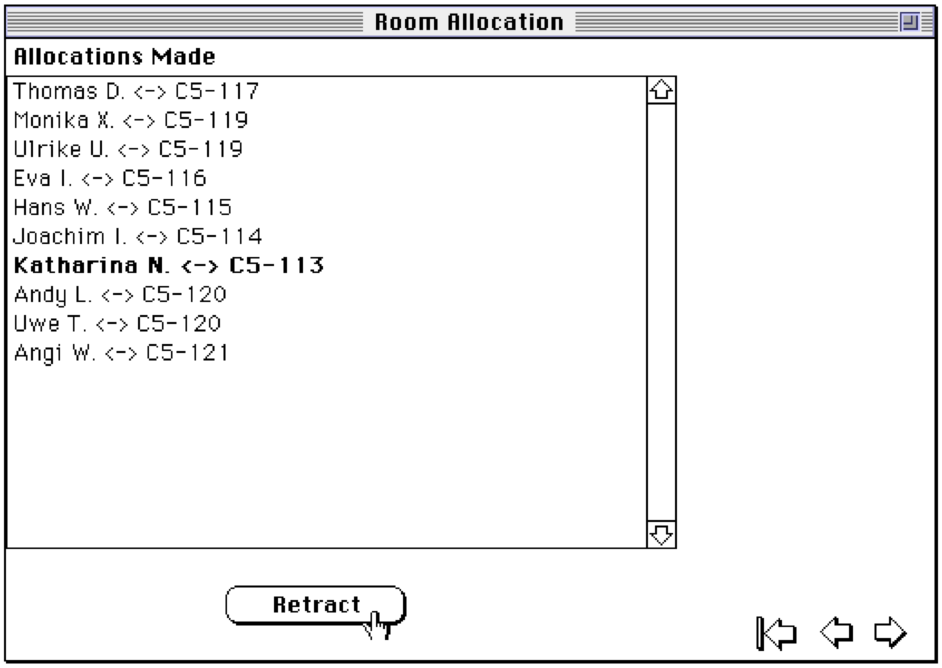


Figure 20 Researcher fill with person on same project omitted

The user can also go to the screen shown in Figure 21 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.



```
Thomas D. <-> C5-117
Monika X. <-> C5-119
Ulrike U. <-> C5-119
Eva I. <-> C5-116
Hans W. <-> C5-115
Joachim I. <-> C5-114
Katharina N. <-> C5-113
Andy L. <-> C5-120
Uwe T. <-> C5-120
Angi W. <-> C5-121
```

Figure 21 Allocations and retraction

The system as described has trouble if Thomas D. is 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 not asserting Thomas D. to also be a member of EULISP in Figure 7. That is, the general question is does one want simple, understandable rules and human intervention in problem posing, or complex rules that take into account the combinatorial complexities possible in the world? There is clearly no universal answer, but it is an interesting and subtle design issue raised by even this fairly simple problem.

7 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 3 through 16, is:

Recommendation 1
Highest priority agenda item: group head task
Person recommended: Thomas D.
Office recommended: C5-117, C5-119
Assignment: Thomas D. to C5-117 (first pair out of two possibilities)

Recommendation 2
Highest priority agenda item: secretarial task
Person recommended: Monika X., Ulrike U.
Office recommended: C5-119
Assignment: Monika X. to C5-119 (first pair out of two possibilities)

Recommendation 3
Highest priority agenda item: secretary fill task
Person recommended: Ulrike U.
Office recommended: C5-119
Assignment: Ulrike U. to C5-119 (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: Hans W., Joachim I., Katharina N.
Office recommended: C5-115
Assignment: Hans W. to C5-115 (first pair out of three possibilities)

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

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

Recommendation 8
Highest priority agenda item: smoking researcher task
Person recommended: Andy L., Uwe T.
Office recommended: C5-120, C5-121, C5-122, C5-123
Assignment: Andy L. to C5-120 (first pair out of eight possibilities)

Recommendation 9
Highest priority agenda item: smoking researcher fill task
Person recommended: Uwe T.
Office recommended: C5-120
Assignment: Uwe T. to C5-120 (only possibility)

Recommendation 10
Highest priority agenda item: non smoking researcher task
Person recommended: Angi W., Harry C., Jürgen L., Marc M., Michael T., Werner, L.
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., Michael T.
Office recommended: C5-121

Assignment: Harry C. to C5-121 (first pair out of four possibilities)
Note: Werner, L. is not recommended—he is a member of the same project as Angi W.

Recommendation 12
Highest priority agenda item: non smoking researcher task
Person recommended: Jürgen L., Marc M., Michael T., Werner, L.
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., Michael T., Werner, L.
Office recommended: C5-122
Assignment: Marc M. to C5-122 (first pair out of three possibilities)

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

Recommendation 15
Highest priority agenda item: non smoking researcher fill task
Person recommended: Werner, L.
Office recommended: C5-123
Assignment: Werner, 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 Werner L. were smoking) the sub-optimal possibilities open would also be apparent in the data generated.

8 THE SECOND PROBLEM

The second problem is one in which the head of MLT, “Katharina N.”, has left and a smoker, “Christian I.”, has joined as a member of the “MLT Project”. This gives a new organization chart as shown in Figure 22, and with “Christian I.”, replacing “Katharina N” in Figures 8 and 10. A simple substitution of Christian for Katharina in room CS-113 is not possible within the rules because Christian is a “researcher” and hence not entitled to the single room which Katharina had as a “project leader”. However, one would hope that the system could make the availability of this simple possibility apparent, even though it violates the rules and hence should not be a recommendation.

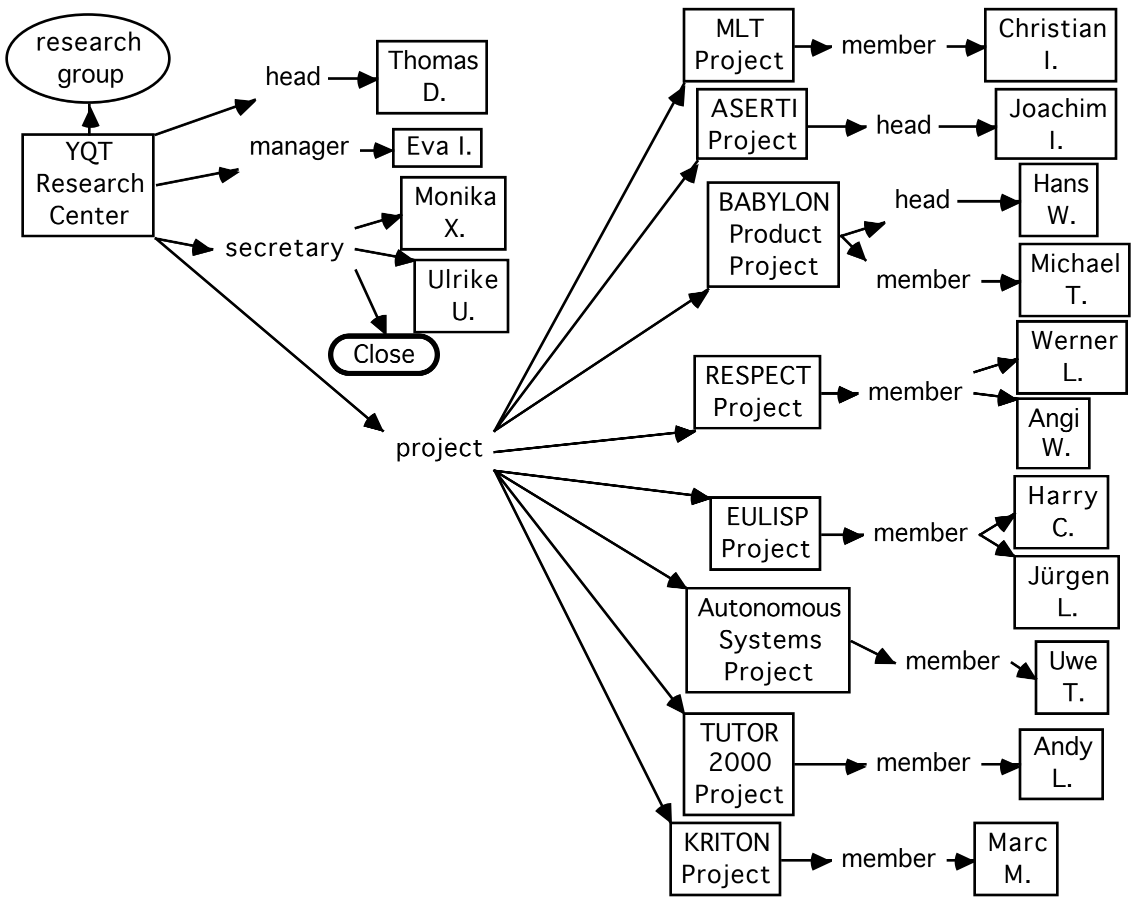


Figure 22 Second problem organization chart

If the problem is run completely again, with the top choices in alphabetic order being taken, it results in the same allocations as before except that room "CS-113" is unused and "Andy L." is paired with "Christian I." in room "CS-120".

The final recommendation is now that shown in Figure 23. No recommendation is available from the rules but, as is apparent in the right hand fields, "Uwe T." has no office and "C5-113" is available so it is simple to allocate him that office. The second problem has no solution within the expert's stated requirements so this is the best that can be done.

If the problem is run incrementally with the allocation of "CS-113" to "Katharina N" retracted, and with "Christian I." added, it results in the screen shown in Figure 23 with "Christian I." replacing "Uwe T.", and hence Christian being allocated Katharina's old office which is a minimally disruptive solution.

Thus, in both the absolute and incremental approaches to this example, the system is able to support the user in making reasonable allocations in situations which are not strictly soluble within the requirements stated in the protocol.

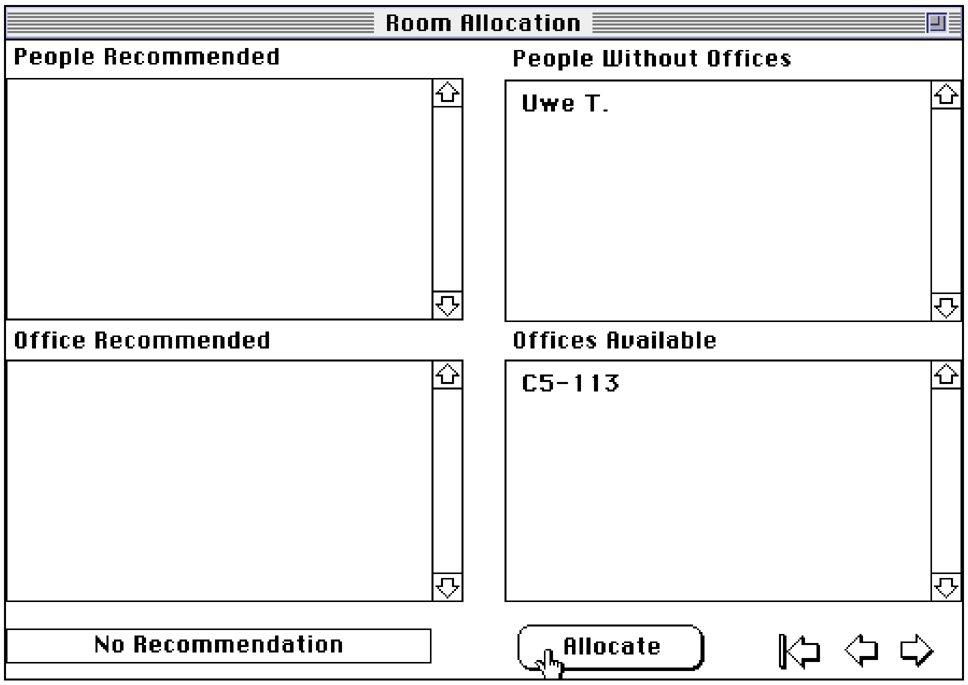


Figure 23 Final recommendation for second problem

9 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 24 shows the generic task structure that may be abstracted. It involves a three way classification of requirements, resources, and tasks resulting from matching classified requirements and resources. The requirements and resources arise from the problem specification, and the tasks arise from the form of solution adopted. A term subsumption knowledge representation schema is used for the representation of the requirements and resources domains, the agenda-based solution mechanism, and the classification rules for requirements, resources, and tasks.

The classification of requirements and resources results in the matching of resources to requirements in tasks on the agenda. The tasks are prioritized as part of the solution specification. Problem requirements that involved further constraints on requirements and resources linked to the same agenda item are implemented through task classification.

The top priority item on the agenda is presented as a recommendation to the user who makes a selection from the recommendations if the problem is under-determined, and makes a selection outside the rules if the problem is over-determined.

The assignments of resources to requirements are tracked so that the final, or partial, solution can be evaluated and assignments retracted to support backtracking if appropriate. This can be used to exhaustively search all possible solutions if required.

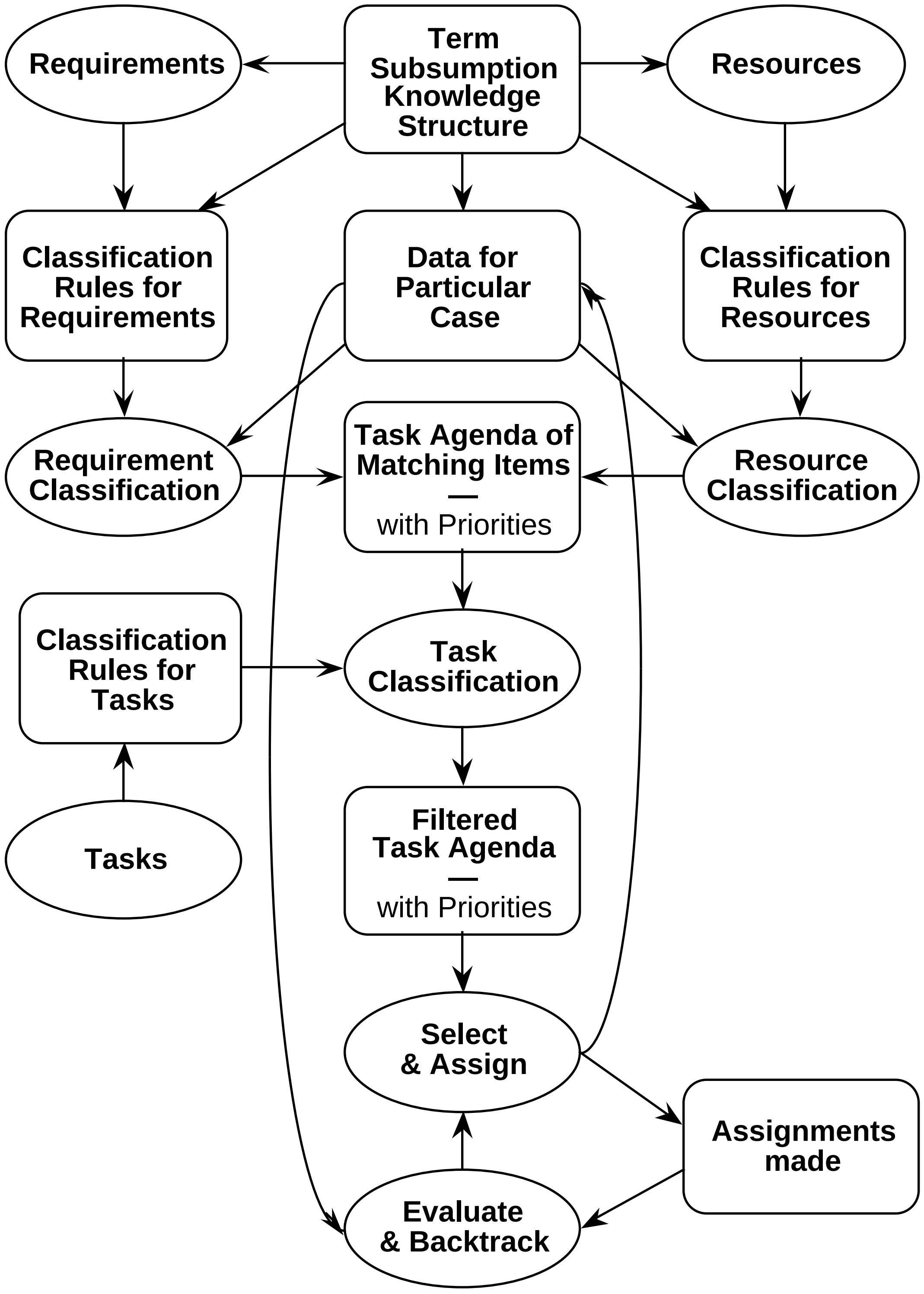

Term Subsumption Knowledge Structure
Requirements
Resources
Classification Rules for Requirements
Data for Particular Case
Classification Rules for Resources
Requirement Classification
Task Agenda of Matching Items — with Priorities
Resource Classification
Classification Rules for Tasks
Task Classification
Tasks
Filtered Task Agenda — with Priorities
Select & Assign
Assignments made
Evaluate & Backtrack

Figure 24 Generic task structure for resource allocation

The knowledge structures that would be edited for variants on the problem are apparent. in Figure 24. 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.

10 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 *situated classification* solution described in this paper 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.

ACKNOWLEDGMENTS

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 grateful to Marc Linster, the GMD and the Esprit *Reflect* Project for making this dataset available through Sisyphus. I am particularly grateful to Mildred Shaw for research collaboration that has shaped many of the ideas reported in this paper, and for comments that greatly improved the clarity of this paper.

REFERENCES

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

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

Clancey, W.J. (1993). The knowledge level reinterpreted: modeling socio-technical systems. **International Journal of Intelligent Systems 8**(1) 33-49.

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

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. San Mateo, California, Morgan Kaufmann.

Gaines, B.R. (1990). Knowledge representation servers: a generic technology for knowledge acquisition and knowledge-based systems. Motoda, H., Mizoguchi, R., Boose, J. and Gaines, B., Ed. **Knowledge Acquisition for Knowledge-Based Systems**. pp.413-430. Tokyo, Ohmsha.

Gaines, B.R. (1991a). Empirical investigations of knowledge representation servers: Design issues and applications experience with KRS. **ACM SIGART Bulletin 2**(3) 45-56.

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.

Gaines, B.R. (1991c). An interactive visual language for term subsumption visual languages. **IJCAI'91: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence**. pp.817-823. San Mateo, California, Morgan Kaufmann.

Gaines, B.R. (1991d). Organizational modeling and problem solving using an object-oriented knowledge representation server and visual language. **COCS'91: Proceedings of Conference on Organizational Computing Systems**. pp.80-94. ACM Press.

Gaines, B.R. (1993). A class library implementation of a principled open architecture knowledge representation server with plug-in data types. **IJCAI'93: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence**. pp.504-509. San Mateo, California, Morgan Kaufmann.

Gaines, B.R. and 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. and Shaw, M.L.G. (1992). Documents as expert systems. Bramer, M.A. and Milne, R.W., Ed. **Research and Development in Expert Systems IX. Proceedings of British Computer Society Expert Systems Conference**. pp.331-349. Cambridge, UK, Cambridge University Press.

Gaines, B.R. and Shaw, M.L.G. (1993a). Basing knowledge acquisition tools in personal construct psychology. **Knowledge Engineering Review 8**(1) 49-85.

Gaines, B.R. and Shaw, M.L.G. (1993b). Eliciting knowledge and transferring it effectively to a knowledge-based systems. **IEEE Transactions on Knowledge and Data Engineering 5**(1) 4-14.

Nosek, J.T. and 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.

Shaw, M.L.G. and Gaines, B.R. (1992). On the relation between the repertory grid and term subsumption knowledge structures: theory, practice and tools. Bramer, M.A. and Milne, R.W., Ed. **Research and Development in Expert Systems IX. Proceedings of British Computer Society Expert Systems Conference**. pp.125-143. Cambridge, UK, Cambridge University Press.

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

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