CommonKADS Models for Knowledge Based Planning [1]

Authors
John Kingston, AIAI, University of Edinburgh
Nigel Shadbolt, AI Group, Department of Psychology, University of Nottingham
Austin Tate, AIAI, University of Edinburgh

AIAI-TR-199

Artificial Intelligence Applications Institute,
University of Edinburgh,
80 South Bridge,
Edinburgh, EH1 1HN
United Kingdom

[1] Extended version of an article published in *Proceedings of AAAI-96*

Abstract

The CommonKADS methodology is a collection of structured methods for building knowledge based systems. A key component of CommonKADS is the library of generic inference models which can be applied to tasks of specified types. These generic models can either be used as frameworks for knowledge acquisition, or to verify the completeness of models developed by analysis of the domain. However, the generic models for some task types, such as knowledge-based planning, are not well-developed. Since knowledge-based planning is an important commercial application of Artificial Intelligence, there is a clear need for the development of generic models for planning tasks.

Many of the generic models which currently exist have been derived from modelling of existing AI systems. These models have the strength of proven applicability. There are a number of well-known and well-tried AI planning systems in existence; one of the best known is the Open Planning Architecture (O-Plan). This paper describes the development of a CommonKADS generic inference model for knowledge-based planning tasks, based on the capabilities of the O-Plan system. The paper also briefly describes the verification of this model in the context of a real-life planning task: the assignment and management of RAF Search and Rescue operations. [2]

Keywords: Knowledge Representation, Expert Systems Design, Planning and Scheduling

1 Introduction

The CommonKADS methodology [WVSA92] [Bv94] is a collection of structured methods for building knowledge based systems. CommonKADS views the development of a knowledge based system as a modelling activity, and so the heart of these methods is the construction of a number of models which represent different views on problem solving behaviour. The CommonKADS methods for developing knowledge-based systems have proved their usefulness repeatedly over a range of different tasks (see [BK92] [Kin92] [LFS93] [Kin93a] for examples).

The key element in the success of CommonKADS is the library of generic inference models which can be applied to tasks of specified types. These models suggest the inference steps which take place in a typical task of that type, and the roles which are played by domain

[2] *Acknowledgements: The work described in this paper was carried out for a project commissioned by the Defence Research Agency Flight Systems Division, Farnborough. The project was entitled "Acquiring and Using Planning Knowledge for Search and Rescue". Thanks are due to the RAF Rescue Co-ordination Centre, Edinburgh for their co-operation and support, in particular Squadron Leader W. Gault. The contributions of the other staff on this project – H. Cottam (University of Nottingham), H. Beck and T. Duncan (AIAI) – are gratefully acknowledged.*

One of the authors was sponsored by the Advanced Projects Research Agency (ARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under grant number F30602-95-1-0022. The U.S. Government is authorised to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ARPA, Rome Laboratory or the U.S. Government.

knowledge in the problem solving process. For example, the generic model for a systematic diagnostic task (e.g. [Kin93b]) includes inference steps such as **decomposing** a set of possible faults, and **matching** observed values against expected values. This model also shows that the set of possible faults plays two roles in the diagnostic process; firstly as a part of a model of the behaviour of a faulty system, and secondly as hypothesised causes of the symptom(s) currently being observed. These generic models can either be used in a top-down manner, as frameworks for knowledge acquisition (e.g. [Kin91]), or they can be used to verify the completeness of models developed bottom-up by analysis of the domain (e.g. [CSK+95]).

CommonKADS' generic model for diagnostic tasks is well-developed and well-understood. However, the generic models for some task types are not as well developed. This is true for tasks involving knowledge-based planning; while CommonKADS does give some guidance in this area [VL94], this guidance focuses on domain models, rather than inference models. Since knowledge-based planning is an important commercial application of Artificial Intelligence, there is a clear need for the development of generic models for planning tasks.

Many of the generic models which currently exist have been derived from existing AI systems, whose operation has been modelled and purged of their domain content. These models have the strength of proven applicability. There are a number of well-known and well-tried AI planning systems in existence; one of the best known is the Open Planning Architecture (O-Plan) [TDK95]. O-Plan, which was developed by AIAI's Knowledge Based Planning and Scheduling Group, provides a generic domain independent computational architecture suitable for command, planning and execution applications. O-Plan makes use of a variety of AI planning techniques, including a hierarchical planning system which can produce plans as partial orders on actions (cf. [Sac77]); an agenda-based control architecture; incremental development of "plan states"; temporal and resource constraint handling (cf. [Ver83]); and a number of data structures used in Nonlin [Tat77] which was the forerunner of O-Plan. It therefore seemed that there would be considerable benefit in using O-Plan as a basis for generating a CommonKADS generic model for planning tasks.

The purpose of this paper is to describe the CommonKADS models which were developed from O-Plan. The paper also briefly describes the verification of these models in the context of a real-life planning task: the assignment and management of Search and Rescue operations by the Royal Air Force.

The format of the paper is:

- A brief description of the CommonKADS methodology;
- A brief description of O-Plan, and how its components relate to the CommonKADS view of knowledge representation;
- A description of the key planning models which were derived from O-Plan;
- A description of how these generic models were verified during the development of a KBS which supported Search and Rescue planning.

2 Knowledge Representation in CommonKADS

CommonKADS is the name of the methodology developed by the KADS-II project, which was funded under the CEC ESPRIT programme [SWd+94]. It is a collection of structured methods for building knowledge based systems, analagous to methods such as SSADM for software engineering. CommonKADS views the construction of KBS as a modelling activity, and so these methods require a number of models to be constructed which represent different views on problem solving behaviour, in its organisational and application context. CommonKADS recommends the construction of six models:

- A model of the organisational function and structure. The key elements of this model are *business processes*, *structural units*, *business resources* and the various *relationships* between them.
- A model of the tasks required to perform a particular operation. The key elements in this model are the *tasks* required for a single business process, and the *assignment* of tasks to various agents.
- A model of the capabilities required of the agents who perform that operation. The key elements of this model are *agents* (human or automated) and their *capabilities.*
- A model of the communication required between agents during the operation. The key elements of this model are *transactions.*
- A model of the expertise required to perform the operation (see below).
- a model of the design of a KBS to perform all or part of this operation. The key step in a CommonKADS design model is (usually) a *functional decomposition* of a knowledge-based process into its component *functional units.*

The key model – the *expertise model* – is divided into three "levels" representing different viewpoints on the expert knowledge:

- The **domain knowledge** which represents the declarative knowledge in the knowledge base. The key elements in domain knowledge are *concepts*, *properties* of concepts, and *relations*. *Tasks* can also be considered to be part of the domain knowledge in some circumstances.
- The **inference knowledge** which represents the knowledge-based inferences which are performed during problem solving. Inference knowledge is represented using *inference functions* (inferences which must be made in the course of problem solving) and *knowledge roles* (domain knowledge which forms the input and output of the inference functions).
- The **task knowledge** which defines a procedural ordering on the inferences. The key elements at this level are *tasks* and their decomposition; in this respect, this level is very similar to the CommonKADS *task model.*

The contents of these three levels can be defined graphically, or using CommonKADS' Conceptual Modelling Language [Bv94] [dHMW+93] [SWAdV94]. For a worked example of the development of each of these three levels, see [Kin93b].

CommonKADS models are typically developed concurrently with the acquisition of knowledge; initial knowledge acquisition is used to populate higher level models (e.g. the organisational or task models) and then these models may be used to document, structure, or guide knowledge acquisition. Partially completed models and/or generic models may even be presented to the experts to allow them to comment on the appropriateness of the models; this technique is similar to the "rapid prototyping" (iterative refinement) approach which was popular in the early days of KBS development. The key difference is that the CommonKADS models are being iteratively refined, rather than an implemented system; this removes many of the problems which were associated with "rapid prototyping" of a KBS, such as lack of documentation, and difficulties in identifying and justifying design decisions.

For more details on the contents of all the models described above, see [dHMW+93].

3 O-Plan: The Open Planning Architecture

The development of open planning and scheduling systems seeks to support incremental extension and change, and to facilitate communication between processing agents (both automated and human). The need to support inter-process communication has become apparent from practical experience; unforeseen events or consequences of concurrent activities can have a major effect on planning, and so the role of the human system operator is crucially important. O-Plan has therefore been designed with an agent-oriented architecture in which job assignment, planning and execution are separated [TDK95], and communication between agents is conducted using the same representations that the planner uses. This separation not only introduces flexibility into the planning process, but also fits well with CommonKADS' multi-viewpoint approach to knowledge representation.

O-Plan is a multi-faceted system, and much has been written about its different features (e.g. [TDK94] [CT91] [DKT92]). The main components of O-Plan are:

- Domain information;
- Plan/schedule states;
- Knowledge sources;
- Controller;
- Several support modules, including constraint managers.

The remainder of this section describes how these components relate to the different models proposed by CommonKADS.

3.1 Domain information

The best model in CommonKADS for representing domain information is the domain level of the expertise model. This model normally contains declarative information about physical objects, states which objects can be in, and relationships between objects; objects and states are represented using *concepts* and *properties*, while relationships are represented by *relations*. However, domain information in O-Plan includes a description of the *activities* which can be undertaken to achieve various planning tasks, as well as information on physical resources available to the planning process (e.g. helicopters, lifeboats, hospitals), and possible states of those resources. The need to represent activities in the domain information implies that the corresponding CommonKADS domain knowledge will include many *tasks* - procedures which can or must be carried out as part of a plan to achieve an objective.[3] From this, it becomes clear that a key factor in knowledge-based planning is the ability to represent activities in a declarative form, so that these activities can be reasoned about. Using this paradigm, the constraints between activities can be represented as relationships between tasks in the CommonKADS domain model.

3.2 Plan states

Plan states have three components: a plan agenda, the planning entities, and plan constraints. The agenda consists of *issues* to be resolved, such as getting a resource into a particular state; planning entities typically consist of planned activities which change the state of resources; and plan constraints provide detailed domain information which constrains further planning, such as the availability of resources. If the Search and Rescue planning task (which is described in section 5) is taken as an example, then an issue might be "a helicopter must be present at the site of the operation"; a planning entity might be "scramble helicopter no. 007 immediately"; and a plan constraint might be "helicopter no. 007 only has enough fuel for 2 hours' flying". [4] This tripartite breakdown of plans corresponds to the <I-N-OVA> (issues, nodes and constraints) model described in [Tat95].

All these components map to *knowledge roles* in the inference level of CommonKADS' expertise model; in other words, they consist of domain knowledge which plays a particular role in problem solving. As a reminder, domain knowledge consists of possible activities, physical resources, possible states of those resources, and relationships between resources and states. At the inference level:

- **Issues** consist of one or more resource states (which need to be achieved), and form an input to a particular planning cycle;
- **Planning entities** in the plan consist of activities, and form the output of a planning cycle;

[3] CommonKADS and O-Plan ascribe different meanings to the term *task*. For the purposes of this paper, O-Plan "activities" and CommonKADS "tasks" can be considered to be broadly equivalent.

[4] It is convenient to consider these three components separately when making the comparison with CommonKADS, even though all of these components can be thought of as constraints on future planning.

- **Plan constraints** consist of both the states of physical resources, and of relationships between planned activities. They provide an intermediate input to a planning cycle.

3.3 Knowledge sources

The knowledge sources in O-Plan address specific planning requirements through the application of plan state modification operators. These include expanding an activity into sub-activities; choosing activities to achieve desired domain states; and selecting resources to perform activities.

These knowledge sources map to *inference steps* (in the inference knowledge of the Expertise model) in the CommonKADS framework. The knowledge sources transform the components of the plan state into other components; for example, an issue from the agenda which is expanded is likely to produce new issues. Since the components of the plan state have been identified as CommonKADS knowledge roles, the knowledge sources must correspond to CommonKADS inference steps.

3.4 Controller

Throughout the plan generation process, O-Plan identifies outstanding issues to address; these issues are then posted on an agenda list. The controller computes the context-dependent priority of the agenda items and selects an item for processing. This provides the fundamental opportunism which is inherent in any planning task.

The knowledge used by the controller could be represented in CommonKADS at the *task level* of the Expertise model (with a few extensions to represent opportunism). The task level specifies ordering on the inference level, and also identifies input and output. For O-Plan, the task knowledge performs reasoning which dynamically determines an ordering on the inference knowledge; this is eminently sensible for any task which involves reacting to a dynamically changing situation, such as planning, scheduling, or control tasks.

3.5 Support modules

Support modules, such as database management facilities or context-layered access to the plan state, do not map into CommonKADS knowledge representation; they are either considered as external agents or extra requirements which have to be considered when the CommonKADS Design model is produced. However, some support modules in O-Plan, such as the constraint managers (which track the availability of resources, the temporal constraints on activities, and the relational constraints on objects), have a considerable effect on the planning cycle. The constraints themselves can be represented as *knowledge roles* in the inference knowledge of the Expertise model.

4 Generic CommonKADS models for Planning

It can be seen from the section above that the knowledge representation structure used in O-Plan corresponds fairly closely with the knowledge representation framework used by CommonKADS; specifically, by the CommonKADS Expertise Model. This made it possible to subdivide the next task in this project, which was to derive generic CommonKADS models for planning from the architecture of O-Plan. It was decided to focus on deriving generic inference models ("inference structures") for the inference level of the CommonKADS Expertise model, since, as noted in section 1, these models often provide most assistance to a KBS developer.[5]

The derived inference structure can be seen in Figure 1. A typical "run" through the inference structure would see the following operations taking place:

- The **current plan state** is notionally decomposed into three components: the **agenda** of issues which are to be resolved, the **current plan entities** and the **constraints**. This decomposition does not alter any of these structures; it simply makes explicit the role which each component of the plan state plays in the problem solving process. These roles are described in [TDK94].
- From the agenda of issues, at least one **issue** is selected for resolution. The choice of an issue depends on a number of factors monitored by the **controller**, such as the available processing capabilities, the knock-on effect on other issues, etc.
- Pattern matching between issues and possible activities is used to find an activity which is capable of resolving the current issue, perhaps by adding entities to the plan, or by creating new issues. Issues may be resolved in one of three ways, which are shown in Figures 2 to 4 below. The "double ellipse" informs the reader that this inference step can be viewed in more detail in other diagrams, in which the uses of the inputs and origins of the outputs are specified more precisely.
- The resulting agenda of issues, plan entities and constraints are assembled, and used to update the current plan.

Figures 2 to 4 show three of O-Plan's "knowledge sources", represented as CommonKADS inference structures. These knowledge sources are each capable of resolving an outstanding issue, but in different ways. The methods used are:

- Adding a new activity, or further constraints on currently planned activities, in order to resolve the issue (Figure 2);

[5]O-Plan can be used for a variety of tasks, including but not limited to planning. For the sake of the current project, it is useful to specify an inference structure which represents the operation of O-Plan as a planner. This inference structure is designed to make explicit the processes which O-Plan goes through when performing planning tasks.

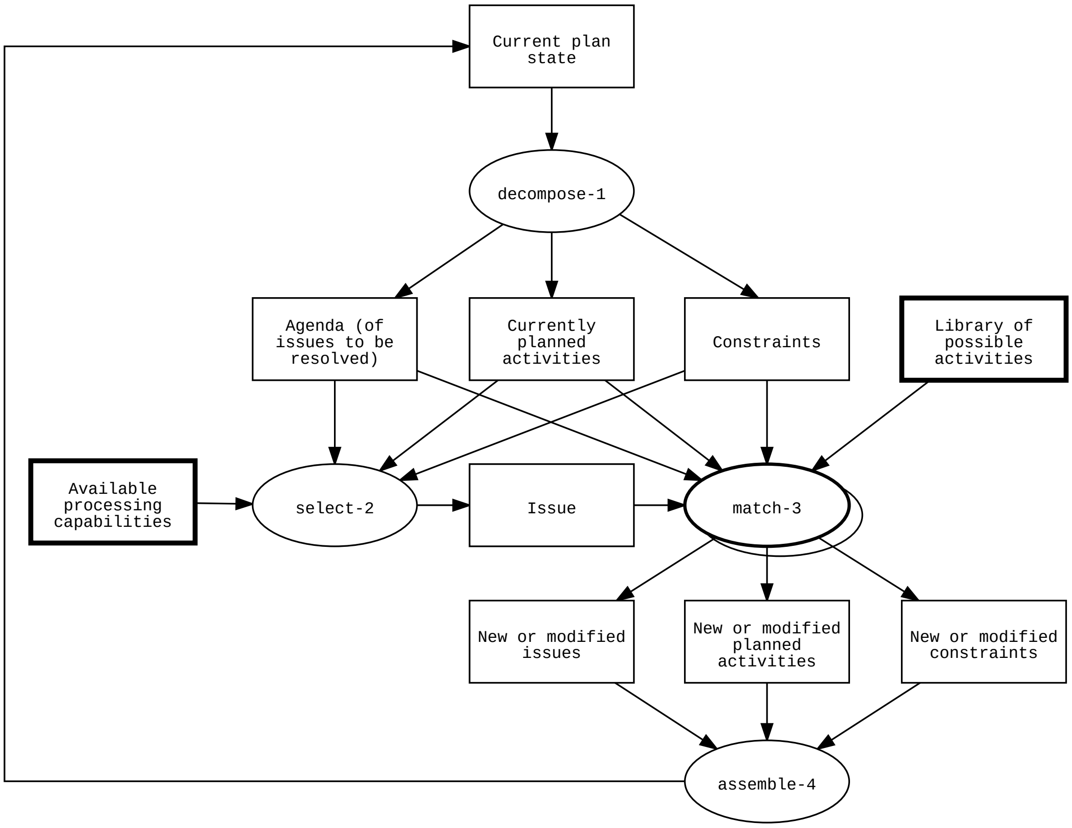

Figure 1: Top level inference structure for the O-Plan planner

- "Backward chaining": adding new issues to the plan which, if resolved, will allow the current issue to be resolved (Figure 3);
- Expanding the issue into a number of sub-issues (Figure 4).

In CommonKADS terms, these three knowledge sources constitute different possible decompositions of the **match-3** inference step. The three decompositions are described in more detail in the following paragraphs.

Figure 2 represents the resolution of an issue by **condition satisfaction**: i.e. the conditions for an issue to be fulfilled are found to be matched. Conditions typically consist of one or more resources being in one or more states. For example, if an issue in the plan was to arrange transport for a mountain rescue team from Kinloss to Ben Nevis, then one possible activity (discovered by **match-3.1.5**) might be to transport the team by helicopter. The conditions of this activity might be that the mountain rescue team is present at a helicopter landing site, and a helicopter is also present at that site; resource constraints and currently planned activities will determine if these conditions can be fulfilled (**match-3.1.6**). If the conditions of an issue are fulfilled, and that issue is selected as the best method of transporting the team (**select-3.1.7**), then that issue is removed from the agenda. The plan itself is also modified, in any or all of the following ways:

- New planning entities may be created (e.g. "helicopter no. 007 must land at Kinloss");
- New variable restrictions may be enforced (e.g. "the helicopter must use the backup landing site at Kinloss");
- New temporal orderings may be introduced (e.g. "the helicopter has to refuel; this must be done before flying to Kinloss").

It is possible that there may be more than one way of matching a set of conditions; for example, there may be more than one helicopter available. In that case, O-Plan automatically selects one option which is used for further depth-first reasoning, and maintains the other options as a choice point in case backtracking is required.

Figure 3 represents the resolution of an issue whose conditions cannot currently be satisfied (as determined by **match-3.2.8**. The approach taken by O-Plan in this case is a form of "backward chaining"; a search is made for other activities which, if added to the plan, will create the right conditions for the current issue to be fulfilled (**match-3.2.9**). If a suitable activity is found, and it is not currently an issue, then the performing of this activity is added to the agenda of issues (**specify-3.2.10**). This is known as **achieving** in O-Plan.

Figure 4 represents the resolution of an issue by **expansion**. If the current issue matches with an activity (**match-3.3.11**) which can be decomposed into sub-activities, then the current issue is removed from the agenda and appropriate sub-issues are created and added to the agenda (**specify-3.3.12**. For example, if "move mountain rescue team to

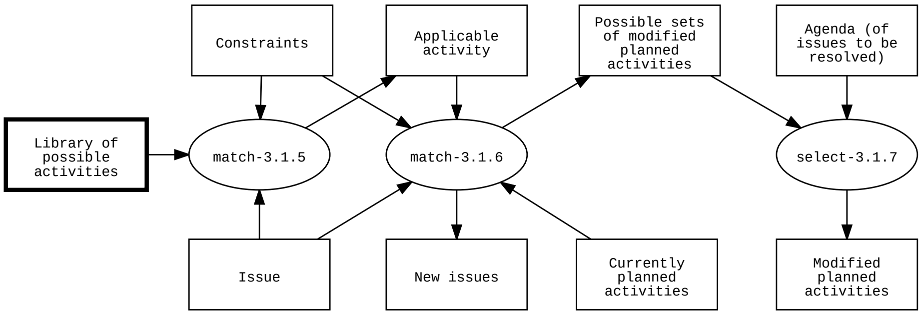

Figure 2: Inference structure for resolving an issue by introducing new activities or constraints into the plan

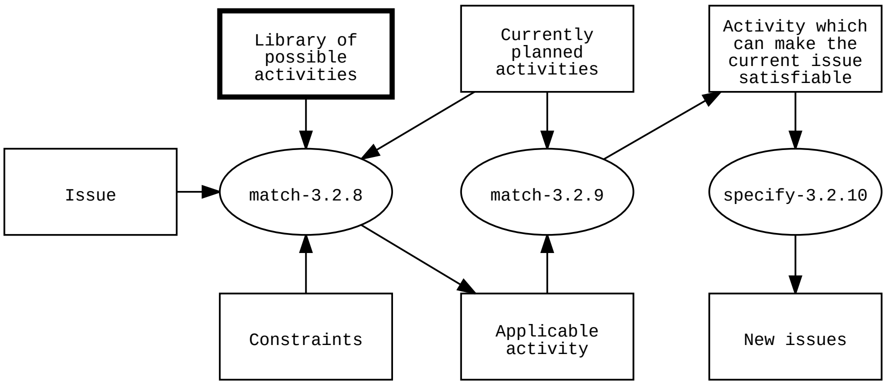

Figure 3: Inference structure for resolving an issue by “backward chaining”

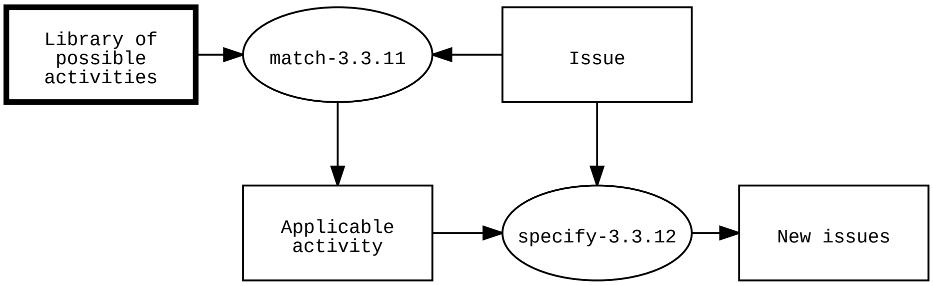

Figure 4: Inference structure for resolving an issue by expanding it into sub-issues

pickup point" was an issue, then this might be expanded into "contact team", "instruct team", and "confirm team have arrived at pickup point".

In summary, these inference structures represent the core activities of the O-Plan planning process, without representing the many controls on efficiency and processing capability which are implemented within the O-Plan Controller; these belong in the task level of the CommonKADS Expertise Model.[6] The system-independence of these inference structures allows them to be used as generic models of the inference processes required for knowledge-based planning.

5 Verifying the generic planning models in the context of Search and Rescue planning

In the previous section, a set of inference structures were derived from the O-Plan approach to planning, and were proposed as generic inference models for knowledge-based planning tasks. Despite the fact that O-Plan is intended to be a generic architecture for implementing different types of knowledge-based planning systems, this proposition is a strong one, because there is a wide variation in task types which fall under the category of knowledge-based planning. Knowledge-based planning tasks may vary in the type of feedback data which is available to the planner[7] [Val94]; in the depth of search required; and in the type of support which a human user needs (fully automated planning vs. monitoring and support of human planning).

5.1 Inference modelling for Search and Rescue planning

In order to verify the claim that the inference structure presented in the previous section can act as a generic inference model for planning tasks, it is therefore important that these models should be seen to be appropriate for real-life planning tasks. One such task is that of planning the use of resources in a Search and Rescue incident. A project entitled "Acquiring and Using Planning Knowledge for Search and Rescue" [CSK+95] was carried out jointly by the University of Nottingham and AIAI, and produced a prototype KBS for supporting Royal Air Force (RAF) personnel in their allocation and management of resources such as Search and Rescue helicopters, RAF mountain rescue teams, and RAF Nimrod aircraft. The responsibilities of the Rescue Co-ordination Centres of the RAF include support and co-ordination of civilian emergencies; this includes direct responsibility for the allocation, application and co-ordination of military resources, as well as co-ordination with a number of civilian emergency authorities such as fire, police, ambulance, coastguard and civilian mountain rescue teams. A rescue incident can

[6]For CommonKADS purists, it should be noted that the detailed information associated with the O-Plan Controller belongs in the task level of the Expertise Model **and** in the problem-solving knowledge which was known as "strategic" knowledge in KADS-I. Since no attempt has yet been made to model the operation of the Controller in CommonKADS, this distinction has not yet been made explicitly.

[7]Valente classifies planners as *linear*, *non-linear*, *reflective* or *skeletal* according to the use which they make of state change data and plan assessment knowledge.

vary in scale from retrieving a walker with a sprained ankle to handling a large aircrash; the Rescue Co-ordination Centres may have to manage several incidents simultaneously, each requiring one or two aircraft as well as one or more other search teams or emergency services.

Knowledge acquisition and high-level task modelling for this system are described in [CSK+95]; the result of these activities was to design and develop a system which supported RAF personnel in making planning decisions, in remembering all the tasks which needed to be undertaken, in deciding what to do next, and in logging actions taken. The system was *not* designed to be a 'closed-loop' planner, which would generate a complete plan with little user consultation; during knowledge acquisition, it was noted that the users always maintained control over the planning process, to the extent that planning is sometimes deliberately delayed until more domain information has been obtained. If the generic inference models which were derived from O-Plan can be shown to be applicable to a system which, unlike O-Plan, is not a closed-loop planner, then the generic models should be applicable to a wide range of knowledge-based planning tasks.

The approach which was taken to the design of the KBS for search and rescue support was to develop a domain-specific inference structure in a bottom-up fashion based on structured interviews, video tape analysis, protocol analysis, incident documentation and structured analysis of specific incident cases [CSK+95]. This inference structure can be seen in Figure 5. Although Figure 5 looks very different from Figures 1-4 at first sight (partly because it uses the terms "goal" and "action" instead of "issue" and "activity"), there are some common components between the two. Figure 5 shows that planning for Search and Rescue operations takes place by choosing an appropriate "template plan", which contains a list of goals (issues) to be satisfied; selecting one of these goals; either matching the goal to an action, or expanding it into a set of sub-goals, which are then individually matched against actions; and then adding all the actions into the current plan.

The generic inference structure was then used to critique the domain-specific inference structure. The result of the comparison showed that the inference structure derived from O-Plan:

- had a richer representation of techniques for matching issues to activities (**match-1** in Figure 5 is replaced by the whole of Figure 2; **decompose** and **match-2** in Figure 5 are replaced by Figure 4; and there is no representation in Figure 5 of the "achieving" represented in Figure 3);
- identified some important knowledge roles (resource constraints, and the library of possible activities) which were not explicitly represented in the domain-driven inference structure

while the domain-derived inference structure highlighted knowledge which is particularly important in the Search and Rescue domain. This primarily consisted of the use of an outline plan template as a framework for planning.

The next stage of modelling is to determine whether the model components which are present in the generic model but do not appear in the domain-derived model are in fact applicable to this planning task. It was easy to determine that the task of Search and Rescue planning is sometimes constrained by available resources (there are only a few helicopters and aircraft available to them), and that the planners select from a library of possible activities when deciding how to fulfil an issue (this is most noticeable when different ways of transporting a casualty to safety are considered). Further investigation also determined that there was (occasionally) a requirement to "achieve" a state of affairs by introducing other activities earlier in the plan. This often occurs when the planners want to use facilities controlled by other authorities, such as lifeboats, which are usually controlled by the Coastguard; in these situations, the facilities cannot be used until permission has been granted by the controlling authority. The issue of "scramble lifeboats" therefore requires the issue of "obtain permission" to be resolved before its conditions can be fulfilled.

The system which was constructed was therefore based on an inference structure which incorporated the best of both worlds; it had all the matching capabilities and inputs of the generic inference structure, as well as the selection of a "template plan" specified by the domain-derived inference structure. The structure of the system was based on the inference structure (with additional transformations and design decisions made using the CommonKADS Design Model); the reasoning component of the system consisted of a number of objects representing possible activities, another set of objects representing issues on the agenda, and a set of rules which matched issues against possible activities. The system also used objects to represent resources (helicopters, mountain rescue teams, etc), and to represent the plan itself, with relations between objects specifying the order of planned activities. User interfaces included a PERT chart-style viewer of the planned activities, a TO DO list showing issues on the agenda, and a "status board" showing the current commitments of resources. For further details, see [CSK+95].

The conclusion which can be drawn is that the generic inference models specified in Figures 1-4 are adequate for representing the task of Search and Rescue planning, once a few domain-specific adaptations have been made[8] ; more importantly, the use of a generic inference model acts as a completeness check on acquired procedural knowledge, by prompting a knowledge engineer to consider possible aspects of the planning process which may not have been identified during initial knowledge acquisition.

6 Future work

We have showed that a set of CommonKADS inference models can be derived to represent the workings of the O-Plan system. We have also seen that these models can be beneficially applied to the modelling of a real-life planning task, identifying important aspects of the task which were not immediately obvious from acquired knowledge. We

[8]Such adaptions are a common feature of KBS projects which use CommonKADS (see [LV93], for example).

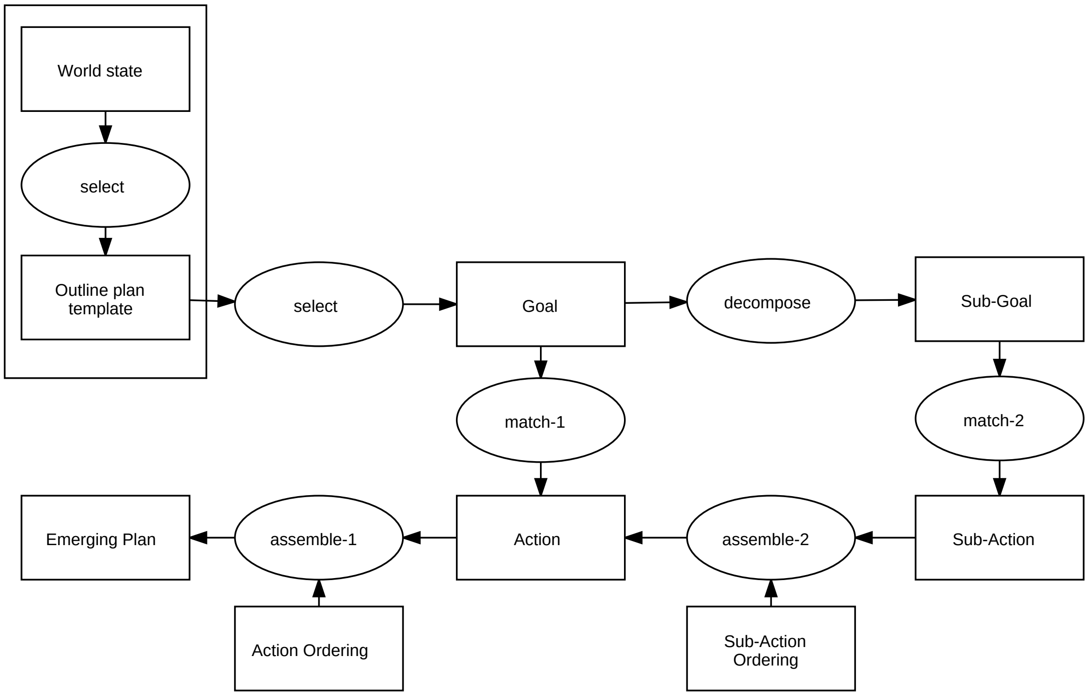

Figure 5: Inference structure derived from knowledge acquisition and domain analysis

can therefore argue that the consideration of these generic models will be beneficial to anyone constructing a planning system, for these models may highlight aspects of the problem which should have been considered.

However, this paper does not claim that the generic inference models highlight *every* aspect that needs to be considered in any planning task. Knowledge-based planning is a wide-ranging field, using a number of different approaches. While O-Plan can perform a wide range of planning tasks (and some other tasks as well), it is based on a particular approach to planning; the inference models derived from O-Plan inevitably reflect the approach. If the generic models shown in Figure 1-4 included control information, then the relationship between O-Plan and the generic models would be the same as the relationship between Mycin and E-Mycin. The deliberate exclusion of control information from CommonKADS inference models helps to lift the generic models to a slightly higher level of abstraction than E-Mycin, but these models cannot be considered equivalent to a generic model for classification tasks, or even for heuristic classification tasks.

What is needed is a top-down approach to classifying planning tasks, which identifies the important characteristics of different approaches to planning, and suggests the types of knowledge which are considered by each type of planning. Since this paper was originally submitted, a paper has been published [BVB96] which takes such an approach, using the CommonKADS framework to produce a high-level description of different planning systems and the approaches which they use. From this perspective, the models produced by Barros *et al* are the "generic" models, specifying the types of operation which a planner is expected to perform (e.g. *select goal* or *critique plan*), whereas the models described in Figures 1-4 are the "domain-derived" models, representing the actual operation of a particular planning system. By applying the same technique of comparing and combining "generic" models with "domain-derived" models, the models described in Figures 1-4 can be verified for completeness, and correctly classified according to the types of planning task for which they are most appropriate, while the models described by Barros *et al* can be enriched. Furthermore, this technique could be used to incorporate a number of other "generic planning models" which have been proposed (such as that of [BC92], and possibly even case-based models such as that used by [GAD+94]) into a common framework, thus permitting rational selection of the "best" generic planning model for a particular planning task.

References

[BC92] D.C. Brown and B. Chandrasekaran. Investigating routine design problem solving. In C. Tong and D. Sriram, editors, *AI in Engineering Design, Vol. 1*. Addison-Wesley, 1992.

[BK92] C. Bauer and W. Karbach. *Proceedings of 2nd KADS User Meeting*. ZFE, Siemens AG, Munich, 1992.

[Bv94] J. Breuker and W. van de Velde. *The CommonKADS Library: reusable components for artificial problem solving*. IOS Press, Amsterdam, Tokyo, 1994.

[BVB96] L. Barros, A. Valente, and R. Benjamins. Modeling planning tasks. In B. Drabble, editor, *Proceedings of 3rd International Conference on AI Planning Systems, AIPS-96*, Edinburgh, Scotland, 29-31 May 1996. AAAI Press.

[CSK+95] H. Cottam, N. Shadbolt, J. Kingston, H. Beck, and A. Tate. Knowledge Level Planning in the Search and Rescue Domain. In M.A. Bramer, J.L. Nealon, and R. Milne, editors, *Research and Development in Expert Systems XII*, pages 309–326. SGES Publications, 11-13 December 1995.

[CT91] K.W. Currie and A Tate. O-Plan: the Open Planning Architecture. *Artificial Intelligence*, 51(1), Autumn 1991. Also available as AIAI-TR-67.

[dHMW+93] R. de Hoog, R. Martil, B. Wielinga, R. Taylor, C. Bright, and W. van de Velde. The Common KADS model set. ESPRIT Project P5248 KADS-II KADS-II/M1/DM1.1b/UvA/018/ 6.0, University of Amsterdam and others, 1993. http://swi.psy.uva.nl/ projects/CommonKADS/Reports.html.

[DKT92] B. Drabble, R.B. Kirby, and A. Tate. O-Plan2: the Open Planning Architecture. In *Working Notes of the AAAI Spring Symposium on Practical Approaches to Scheduling and Planning*, Stanford University, California, 25 - 27 March 1992. American Association for Artificial Intelligence. http://www.aiai.ed.ac.uk/ oplan/oplan/oplan-doc.html.

[GAD+94] A. K. Goel, K. S. Ali, M. W. Donellan, A. G. de Silva Garza, and T. J. Callantine. Multistrategy Adaptive Path Planning. *IEEE Expert*, pages 57–65, Dec 1994.

[Kin91] J.K.C. Kingston. X-MATE: Creating an interpretation model for credit risk assessment. In *Expert Systems 91*. British Computer Society, Cambridge University Press, 17-18 Sep 1991. Also available from AIAI as AIAI-TR-98.

[Kin92] J. K. C. Kingston. KBS Methodology as a framework for Co-operative Working. In *Research and Development in Expert Systems IX*. British Computer Society, Cambridge University Press, 16-17 Dec 1992. Also available from AIAI as AIAI-TR-130.

[Kin93a] J.K.C. Kingston. Pragmatic KADS 1.0. Technical Report AIAI-IR-13, AIAI, University of Edinburgh, 1993.

[Kin93b] J.K.C. Kingston. Re-engineering IMPRESS and X-MATE using CommonKADS. In *Research and Development in Expert Systems X*, pages 17–42. Cambridge University Press, 1993. http://www.aiai.ed.ac.uk/ jkk/publications.html.

[LFS93] C. Löckenhoff, D. Fensel, and R. Studer. *Proceedings of 3rd KADS User Meeting*. ZFE, Siemens AG, Munich, 1993.

[LV93] C. Löckenhoff and A. Valente. A Library of Assessment Modelling Components. In *Proceedings of 3rd European KADS User Group Meeting*, pages 289–303. Siemens, Munich, March 1993.

[Sac77] E.D. Sacerdoti. *A Structure for Plans and Behaviour*. Artificial Intelligence. Elsevier North Holland, 1977.

[SWAdV94] G. Schreiber, B. Wielinga, H. Akkermans, and W. Van de Velde. CML: The CommonKADS Conceptual Modelling Language. In L. Steels, G. Schreiber, and W. van de Velde, editors, *A future for knowledge acquisition: Proceedings of EKAW-94*, Hoegaarden, Belgium, Sept 26-29 1994. Springer-Verlag.

[SWd+94] G. Schreiber, B. Wielinga, R. de Hoog, H. Akkermans, and W. van de Velde. CommonKADS: A Comprehensive Methodology for KBS Development. *IEEE Expert*, pages 28–37, Dec 1994.

[Tat77] A. Tate. Generating project networks. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 888–893, Los Altos California, 1977. William Kaufmann Inc.

[Tat95] A. Tate. Characterising Plans as a Set of Constraints - the <I-N-OVA> Model - a Framework for Comparative Analysis. *ACM SIGART Bulletin: Special Issue on Evaluation of Plans, Planners, and Planning Agents*, 6(1), January 1995. A paper on this subject will also be presented at AIPS-96.

[TDK94] A. Tate, B. Drabble, and R. Kirby. O-Plan2: an Open Architecture for Command, Planning and Control. In M. Zweben and M. Fox, editors, *Intelligent Scheduling*, pages 213–239. Morgan-Kaufmann, 1994.

[TDK95] A. Tate, B. Drabble, and R. Kirby. O-plan2: an open architecture for command, planning and control. Technical report, AIAI, 1995.

[Val94] A. Valente. Planning Models for the CommonKADS Library. In J. Breuker and W. van de Velde, editors, *The CommonKADS Library*, chapter 11. IOS Press, 1994.

[Ver83] S. Vere. Planning in time: Windows and durations for activities and goals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:246–267, 1983.

[VL94] A. Valente and C. Löckenhoff. Assessment. In J. Breuker and W. van de Velde, editors, *The CommonKADS Library*, chapter 8. IOS Press, 1994.

[WVSA92] B. Wielinga, W. Van de Velde, G. Schreiber, and H. Akkermans. The KADS Knowledge Modelling Approach. In *Proceedings of the Japanese Knowledge Acquisition Workshop (JKAW'92)*. 1992.