CommonKADS Models for Knowledge-Based Planning

John Kingston, Nigel Shadbolt*and Austin Tate

AIAI
University of Edinburgh
Edinburgh, Scotland EH1 1HN
J.Kingston@ed.ac.uk

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 describes the verification of this model in the context of a real-life planning task: the assignment and management of Royal Air Force Search and Rescue operations.

Introduction

The CommonKADS methodology (Breuker & van de Velde 1994) is a collection of structured methods for modelling different aspects of knowledge-based systems. These methods have proved their usefulness repeatedly over a range of different tasks (e.g. (Löckenhoff, Fensel, & Studer 1993) (Kingston 1993)). 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 knowledge in the problem solving process. These generic models can be used either in a top-down manner, as frameworks for knowledge acquisition, or to verify the completeness of models developed bottom-up by analysis of the domain.

CommonKADS' generic models for tasks such as diagnosis are 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 (Valente & Löckenhoff 1994), 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) (Tate, Drabble, & Dalton 1996). O-Plan provides a generic domain independent computational architecture suitable for command, planning and execution applications. It makes use of a variety of AI planning techniques, including a hierarchical planning system which can produce plans as partial orders on actions; an agenda-based control architecture; incremental development of "plan states"; and temporal and resource constraint handling. It therefore seemed that there would be considerable benefit in basing a generic CommonKADS model for planning tasks on O-Plan.

CommonKADS

CommonKADS 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 a 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

*AI Group, Department of Psychology, University of Nottingham, University Park, Nottingham NG7 2RD recommends the construction of six models:

- A model of the *organisation's* function & structure.
- A model of the *tasks* (activities) required to perform a particular operation.
- A model of the capabilities required of the *agents* who perform that operation.
- A model of the *communication* required between agents during the operation.
- 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.

For more details on the contents of all the models described above, see (de Hoog *et al.* 1993).

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, often using a semi-formal textual representation. The key elements at this level are *tasks* and their decomposition.

The contents of these three levels can be defined graphically, or using CommonKADS' Conceptual Modelling Language (Schreiber *et al.* 1994); see (Kingston 1993) for a worked example.

Initial knowledge acquisition is typically used to populate higher level models (e.g. the organisational or task model. These models are then used to document, structure, and guide further knowledge acquisition, which is then fed back into more detailed models. To use a familiar analogy, the models are "rapidly prototyped" until they are judged to be adequate for use as the specification of a KBS.

The Open Planning Architecture

O-Plan (Tate, Drabble, & Dalton 1996) is a multifaceted system with an agent-oriented architecture in which task assignment, planning and execution are separated. The main components of an O-Plan agent are:

- Domain information;
- Plan/schedule states;
- Knowledge sources;
- Controller;
- Support modules.

The remainder of this section describes how these components relate to CommonKADS' models.

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). 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.

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. 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. This tripartite breakdown of plans also corresponds to the <I-N-OVA> (issues, nodes and constraints) model described in (Tate 1995).

All these components map to *knowledge roles* in the inference level of CommonKADS' expertise model; that is, they consist of domain knowledge which plays a particular role in problem solving:

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

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 knowledge roles, the knowledge sources must correspond to inference steps.

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 opportunism which is fundamental to any planning task.

In CommonKADS terminology, the controller dynamically determines an ordering on the inference knowledge. The knowledge used by the controller could therefore be represented in CommonKADS at the *task level* of the Expertise model (with a few extensions to represent opportunism).

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, 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.

CommonKADS models for Planning

It can be seen from the section above that the knowledge representation structure used in O-Plan corresponds well with the knowledge representation framework used by CommonKADS. This made it possible to focus on deriving generic inference models ("inference structures") from O-Plan; as noted previously, these models often provide most assistance to a KBS developer.

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

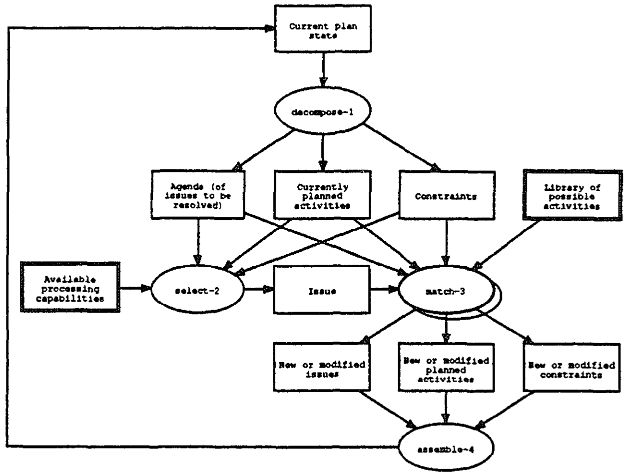

Figure 1: Top level inference structure for O-Plan

- The **current plan state** is notionally decomposed into three components: the **agenda** of issues which are to be resolved, the **currently planned activities** and the **constraints**. This decomposition simply makes explicit the role which each component of the plan state plays in the problem solving process. These roles are described in (Tate, Drabble, & Dalton 1996).
- 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; the "double ellipse" informs the reader that more detail is available in other diagrams (Figures 2 to 4).
- 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);
- "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 below.

Figure 2 represents the resolution of an issue by **condition satisfaction**: i.e. the conditions of an outstanding issue 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 introduced (e.g. "helicopter no. 007 must land at Kinloss");
- New variable restrictions may be enforced (e.g. "the helicopter must have space for 8 men when it arrives at Kinloss");
- New temporal orderings may be introduced (e.g. "the helicopter has to refuel; this must be done before flying to Kinloss").

If there is be more than one way of matching a set of conditions, O-Plan performs search to investigate one or more options.

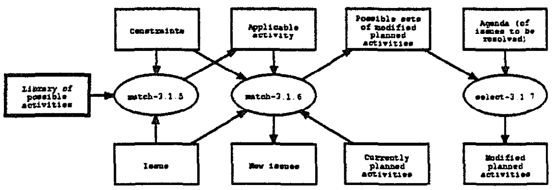

Figure 2: Resolving an issue by condition satisfaction

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, 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

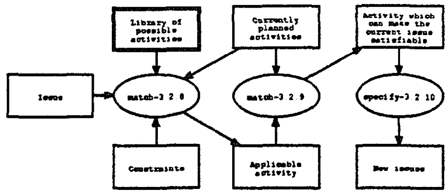

Figure 3: Resolving an issue by "backward chaining"

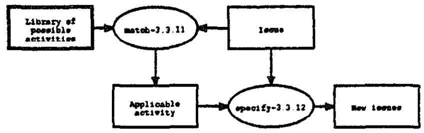

Figure 4: Resolving an issue by expansion

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 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. The system-independence of these inference structures allows them to be used as generic models of the inference processes required for knowledge-based planning.

Verifying the generic planning models

In order to verify the claim that the inference structures presented in the previous section can act as a generic inference model for planning tasks, it was 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" (Cottam *et al.* 1995) 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 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. A rescue incident can 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.

Knowledge acquisition and high-level task modelling for this system are described in (Cottam *et al.* 1995); 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 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 (Cottam *et al.* 1995). This inference structure can be seen in Figure 5. 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 (activity), 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 models were used to critique this domain-specific inference structure. The result of the comparison showed that the generic inference models 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. It also identified some important knowledge roles (resource constraints, and the library of possible activities) which were not explicitly represented in the domain-driven inference structure. On the other hand, the domain-derived inference structure highlighted use of an outline plan template as a framework for planning, which is important in the Search and Rescue domain, but does not appear in the generic inference models.

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), 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 KBS which was implemented was therefore

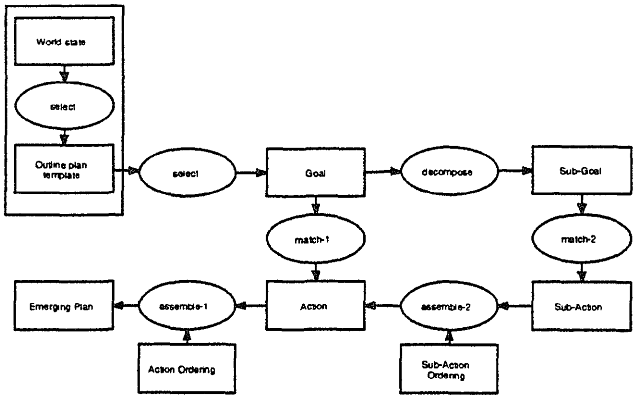

Figure 5: Domain-derived inference structure

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 program structure of the implemented KBS was based on the inference structure (with additional transformations and design decisions made using the CommonKADS Design Model); the system used objects to represent possible activities, another set of objects to represent 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 relations between objects to specify the order of planned activities. For further details, see (Cottam *et al.* 1995).

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; such adaptions are a common feature of KBS projects which use CommonKADS (see (Valente & Löckenhoff 1994), for example). 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.

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 can therefore argue that the consideration of these generic models will be beneficial to anyone constructing a planning system, for the application of 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. The problem is analagous to the abstraction of models from Mycin program; the level of abstraction of the models shown in Figure 1-4 is higher than that of E-Mycin (because of the deliberate exclusion of control information from CommonKADS inference models), and may prove to be comparable with Clancey's model of heuristic classification (Clancey 1985), but cannot be considered to provide a generic model for all 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 work was performed, a paper has been published (Barros, Valente, & Benjamins 1996) 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 techniques described above 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 (Brown & Chandrasekaran 1992), and possibly even case-based models such as that used by (Goel *et al.* 1994)) into a common framework, thus permitting rational selection of the "best" generic planning model for a particular planning task.

Acknowledgements:

The work described in this paper was carried out for a project commissioned by the UK 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 Coordination 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, H. Beck and T. Duncan), and support from the US ARPA/Rome Laboratory Planning Initiative under grant number F30602-95-1-0022 for one author, are gratefully acknowledged. The U.S. Government is authorised to reproduce and distribute reprints for Government use, notwithstanding any copyright notation hereon.

The views and conclusions contained herein are those of the authors 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.

References

Barros, L.; Valente, A.; and Benjamins, R. 1996. Modeling planning tasks. In Drabble, B., ed., *Proceedings of 3rd International Conference on AI Planning Systems, AIPS-96.* Edinburgh, Scotland: AAAI Press.

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

Brown, D., and Chandrasekaran, B. 1992. Investigating routine design problem solving. In Tong, C., and Sriram, D., eds., *AI in Engineering Design.* Addison-Wesley.

Clancey, W. 1985. Heuristic Classification. *Artificial Intelligence* 27.

Cottam, H.; Shadbolt, N.; Kingston, J.; Beck, H.; and Tate, A. 1995. Knowledge Level Planning in the Search and Rescue Domain. In Bramer, M.; Nealon, J.; and Milne, R., eds., *Research and Development in Expert Systems XII*, 309–326. SGES Publications.

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

Goel, A. K.; Ali, K. S.; Donellan, M. W.; de Silva Garza, A. G.; and Callantine, T. J. 1994. Multistrategy Adaptive Path Planning. *IEEE Expert* 57–65.

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

Löckenhoff, C.; Fensel, D.; and Studer, R. 1993. *Proc. of 3rd KADS User Meeting.* ZFE, Siemens AG, Munich.

Schreiber, G.; Wielinga, B.; Akkermans, H.; and de Velde, W. V. 1994. CML: The CommonKADS Conceptual Modelling Language. In Steels, L.; Schreiber, G.; and van de Velde, W., eds., *A future for knowledge acquisition: Proc. of EKAW-94.* Hoegaarden, Belgium: Springer-Verlag.

Tate, A.; Drabble, B.; and Dalton, J. 1996. O-Plan: A knowledge-based planner and its application to logistics. In Tate, A., ed., *Advanced Planning Techniques.* Morgan Kaufmann. http://www.aiai.ed.ac.uk/~oplan/oplan/oplan-doc.html.

Tate, A. 1995. 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). Another paper on this subject will be presented at AIPS-96.

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