CommonKADS and Cooperating Knowledge Based Systems

Hans-Peter Weih, Joachim Schü, Jacques Calmet
Institut für Algorithmen und Kognitive Systeme
University of Karlsruhe
E-Mail: {weih,schue,calmet}@ira.uka.de

March 8, 1994

Abstract

In this paper we shall deal with the knowledge acquisition for cooperating knowledge based systems using a preliminary version of the CommonKADS methodology with minor extensions. The main result of our study is the description of the task, inference and domain knowledge of a general supervisor agent which is responsible for communication and cooperation in our CKBS. According to the CommonKADS philosophy task and inference knowledge of such a general supervisor agent are reusable indepently of the underlying CKBS shell. We will show the applicability of CommonKADS for CKBS Knowledge Acquisition in a macroscopic sense. Particulariliy on the domain layer of the expertise model there is a need to provide a model for reasoning with different models of uncertainty. During the result recomposition stage the supervisor agent has furthermore to deal with inconsistent knowledge due to the different viewpoints that agents in an overlapping application area might have, e.g. different doctors.

1 Introduction

Cooperating Knowledge Based Systems (CKBS) or applicable Multi-Agent Systems were recently established as a new research area amalgamating results from the distributed artificial intelligence (DAI) [1] and distributed database (DDB) communities [2]. In this paper we shall deal with the knowledge acquisition for CKBS using a preliminary version of the CommonKADS[1] methodology. Our work could be considered as an evaluation of the suitability of CommonKADS for CKBS. The main result of our research is the description of the task, inference and domain knowledge of a general supervisor agent which is responsible for communication and cooperation in our CKBS [12]. According to the CommonKADS philosophy, task and inference knowledge of such a general supervisor agent are reusable indepently of the underlying CKBS shell. This paper sketches some of the key ideas which make it possible. It is structured as follows. In the second section we shall give a very short introduction to CommonKADS and its models. In section 3

[1] Final results are expected in 1994

we prove the applicablity of the model set with minor modifications in the agent and organization model for CKBS. We illustrate our work by providing a sample scenario for a hospital. In section 4 we will discuss a former approach, SOM [4] extending KADS for DAI scenarios.

2 CommonKADS

The CommonKADS method is characterized by the models, in which the knowledge is described. To make the paper self-contained a brief overview on the model set is provided [6]:

Organization Model describes the organization in which the knowledge based system should be installed. It is described by organizational problems and organizational descriptors about the function, structure, process, power, authority and resources of an organization.

Task Model : the Task Model describes the tasks that are executed in the organizational environment where the proposed expert system will be installed in the future. The Task Model is represented as a set of tasks with a structure imposed on it. A task is described by its input, output, related ingredients, goal of the task, control, features, environmental constraints and its required capabilities.

Agent Model : in the agent model all relevant properties of agents, capable of solving tasks of the task model are described. An agent might be an expert system, a user, a software system, a database or any other "entity" able to perform a task. An agent is described by ist general capabilities, its constraints and reasoning capabilities.

Communication Model : the communication model describes all transactions that cross the boundaries beetween agents. It describes the communication between the agents in terms of transactions, transaction plans, initiatives and capabilities to take part in a transaction.

Expertise Model : the expertise model describes what capabilities an agent has with a bias towards knowledge intensive problem solving capabilities. The expertise model is divided into domain knowledge, inference knowledge, task knowledge and strategic knowledge as known in KADS. The description is much more detailled than in KADS.

During the model building process, states are introduced to determine whether a model is in developement (transition state) or has already reached a predefined state (landmark state) in the model building process. A quality criteria is introduced to determine whether a landmark status is reached or not.

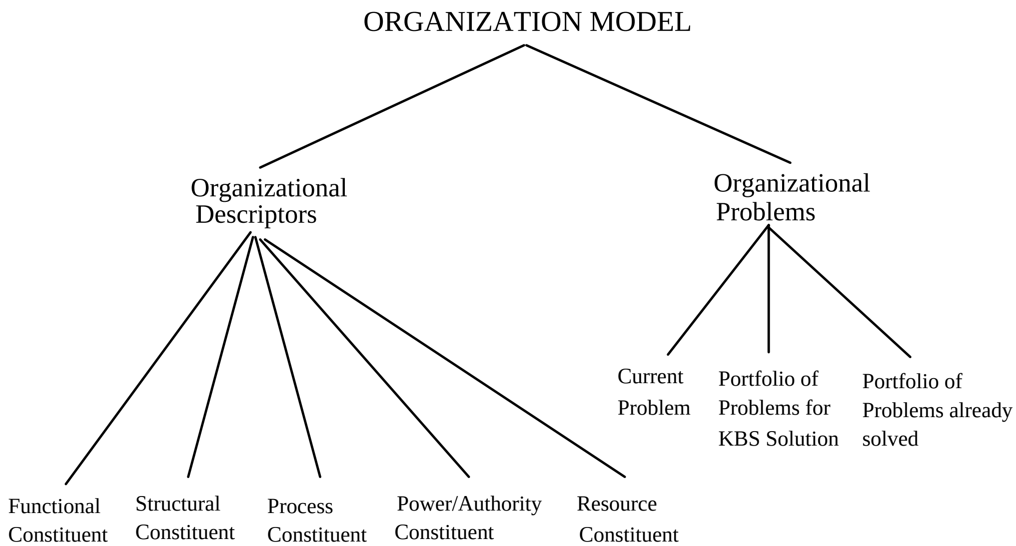

Figure 1: The expanded organization model.

3 CommonKADS and CKBS

In this section we are summarizing our attempt to apply the CommonKADS methodology to CKBS knowledge acquisition. We will give an overview on how to tailor the necessary models and describe some minor modifications.

3.1 Organization Model

In addition to the portfolios described in [6] we introduce a third portfolio which includes all the problems already solved by an expert system, software system or a database system (see figure 1). This will be useful in the task model.

We will now introduce an example which will be furthermore expanded:

A hospital consists of different specialised wards like surgery, internal medicine, paediatrics, neurology etc. All these wards make diagnosis of diseases and give treatments along different methods and in different ways. In each ward there are specialized doctors, nurses and laboratory staff. Most of the time, an illness is not exactly restricted to one field. Many fields in medicine overlap each other. As a result, there is a need of cooperation among the different wards in the hospital in order to make a good diagnosis and give treatments to patients. Expert systems, databases and sensors can support the diagnosis of a doctor, to make him surer of his/her diagnosis. A CKBS could be a solution to support this process connecting all existing expert systems, databases and sensors.

The organization model for such a scenario is as follows:

Current Problem : improvement of diagnosis and therapy.

Portfolio of problems for KBS solution : diagnosis of cancer, allergy etc.

Portfolio of problems already solved : infection with bacteria, diagnosis for the lungs, diagnosis for the heart etc.

Functional : description of the functions of the hospital.

Structural : the segmentation of the hospital into wards.

Resources : describe the personal and material resources of the hospital. E.g. which equipment is available and whose and how many people are doing what job.

Power/Authority : the power and authority structure in the hospital (based on hierarchy and/or knowledge).

3.2 Task Model

In the Task Model we do a task decomposition, which should stop, if there is a subtask already solved by a computer aided system, or if the problem is small enough to be solved easily.

Subtasks which could be solved by a knowledge based system become part of a new CommonKADS product. This product is seen under the environment given by the CKBS. Thus, some of the templates in this CommonKADS product are already predefined.

The subtasks are described by their attributes and functions, as given above. In case of the absense of a current computer aided solution for a particular subtask, the attributes and functions of it are described. In the following process a black box of this subtask is used. There is no necessity to be aware of how the function of the subtask is processed. The tasks are related to each other by the dataflow between the tasks.

The next step is to determine, which subtasks could be executed by which agents, based on the description of the capabilities needed to perform a task and the description of the capabilities of the agents. Afterwards, the negotiation space can be determined; that means all possible assignments of agents to tasks [5]. Based on the negotiation space, all ingredients can be fixed. Ingredients are those pieces of information which are crossing the boundaries between agents. Using these ingredients it will be possible to determine the transactions in the communication model [5].

The result of the task decomposition, assignment of tasks to agents and the determination of ingredients is shown in figure 2. The treatment of a patient in a hospital consists of the diagnosis and the therapy of a patient. The diagnosis is divided into the registration of data, and different specialized branches to diagnose an illness (like infection, heart disease) and the final diagnosis based on the results of all the other tasks. Therapy consists of propositions for the therapy, carrying out of the therapy and the controlling of its results. The results of a therapy can be useful to ensure a diagnosis or to correct a diagnosis. The assignment of agents to tasks is also shown. Tasks can be executed by more than one agent. This leads to horizontal cooperation which must be treated in a special way.

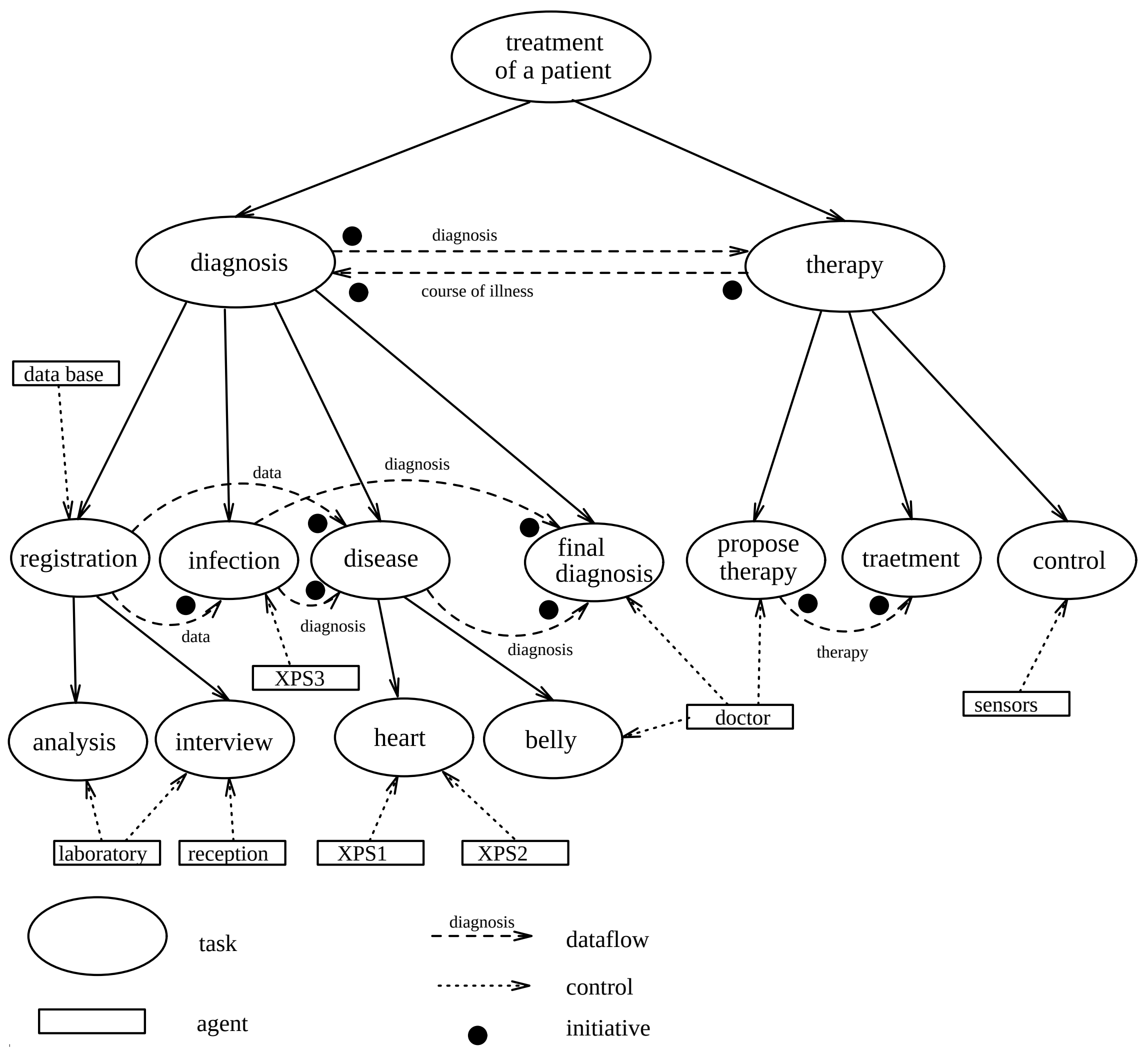

Figure 2: The task decomposition of the hospital problem.

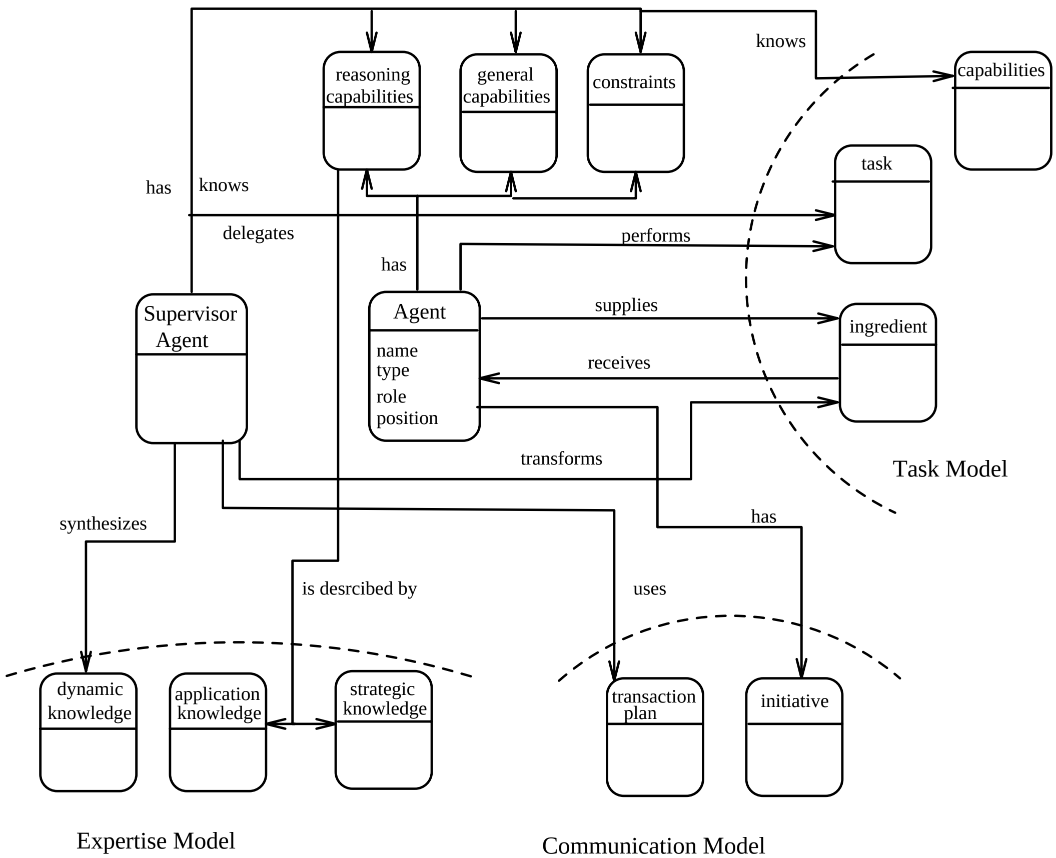

Figure 3: Agent model for CKBS

3.3 Agent Model

In addition to the normal agents we introduce a supervisor agent, which carries out nono of the tasks of the task model, but rather is responsible for the communication and co-operation between the other agents.

There is no difference in using a centralized or decentralized architecture for the future distributed system, since the knowledge of the supervisor agent could be attached to every local expert system shell. The work of the supervisor agent consists of:

- the decomposition of tasks,
- the delegation of tasks,
- the control of the communication,
- the transformation of knowledge representation and
- the synthesis of inconsistent knowledge.

The decomposition of the tasks was already done in the task model. Using the information of the task model and the capabilities of the agents in the agent model, the supervisor can delegate the tasks to the agents. To control the communication, he uses the protocol developed in the communication model. The transformation of knowledge representation and the synthesis of inconsistent knowledge is described in the domain knowledge. Delegation, decomposition and the control of the communication are inference steps described in the inference knowledge.

A formal description of the supervisory agent is given in figure 3. The supervisor agents knows the general capabilities and the reasoning capabilities of the other agents. He uses the transactions and the transaction plan to control the communication. He transforms the ingredients and synthesizes the dynamic knowledge of the CKBS. His reasoning capabilities are described in the expertise model.

3.4 Communication Model

In CKBS the communication model becomes more important than in normal expert systems. The communication between the different agents must be modeled. Agents may have different preferences, goals and knowledge. A scenario can be that several doctors have to set a date for a complicated surgery. Included in the CKBS is also a timetable manager as described in [9]. In figure 4 a protocol shows, how to arrange a date among the different doctors. An inquiry for an arrangement can be rejected, accepted or modified according to the protocol. At the end, after negotiating and refining the dates there is an arrangement or a failure.

3.5 Expertise Model

In the expertise model we will describe the task, inference and domain knowledge based on HECODES, which is a CKBS with one central node for managing the communication and cooperation (cf. [14]). But the knowledge described is independent of the future architecture. If a system has more than one node, each node shares the same information.

3.5.1 Task Knowledge

The task for a distributed system is shown in figure 5. It shows the control over the different inference steps described in the inference knowledge. The supervisory agent receives data, decomposes tasks until no further decomposition is possible, synthesizes solutions from different agents, delegates tasks and transforms the data when needed. Finally, he sends the data to the relevant system.

3.5.2 Inference Knowledge

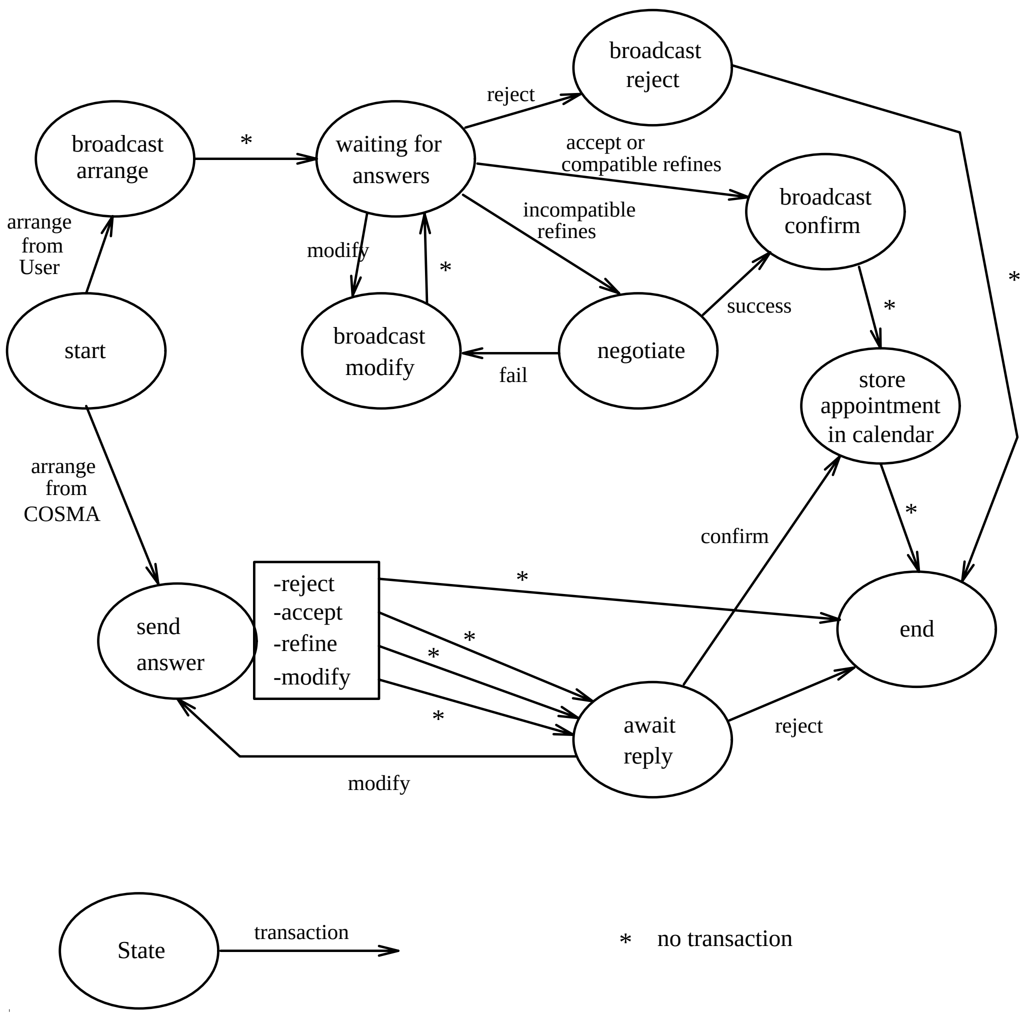

Figure 4: Protocol of the COSMA date planer.

```
loop
  receive (data)
  if (data is a task) then
    while (decomposition is possible)
      decompose (tasks and subtasks)
    end_while
    delegate
  else
    if (horizontal cooperation) then synthesize (solutions)
    end_if
  end_if
  if (representation of the source and destination are different) then transform (solution)
  end_if
  send (data)
end_loop
```

Figure 5: Task knowledge of the supervisor agent.

The dataflow and the inference steps of the inference knowledge is given in figure 6. The idle state, at the top of the figure, characterizes the reactive agent. The supervisor agent expects that other agents or users start an initiative. He receives data according to the protocol. Tasks become decomposed, delegated and transformed, solutions become synthesized and transformed as well. Finally, the data are sent to the destination agent. Two important features are the capabilities of the agents, the protocol and the meta knowledge. Meta knowledge are rules used to solve inconsistent knowledge, to transform the representation of the knowledge and are the informations about the decomposition of the tasks.

3.5.3 Domain Knowledge

In the former sections we stressed the applicabilty of the CommonKADS methodology for the Knowledge Acquisition of CKBS in a macroscopic sense. In this section we focus on microscopic aspects of the inter-agent communication. Due to the different viewpoints different local expert systems might have, there is a need to reason formally with inconsistent knowledge. This supposes that there is a common terminology among the agents. During the result recomposition stage a supervisor agent has to resolve those inconsistencies, e.g. by *prefering* a particular agents opinion. Another task would be to transform different knowledge representation schemes, particulariliy different models for uncertain reasoning.

Let us first have a closer look into the transformation of different models for uncertain reasoning. In a CKBS there might be different representations for the evidence of a diagnosis, e.g. EMYCIN uses the intervall $[-1 \ldots 1]$ to represent the evidence of a diagnosis. -1 represents that the diagnosis is excluded; 0, that nothing is known about the diagnosis and 1, that the diagnosis is true [14]. D3 uses the INTERNIST model [8], which makes

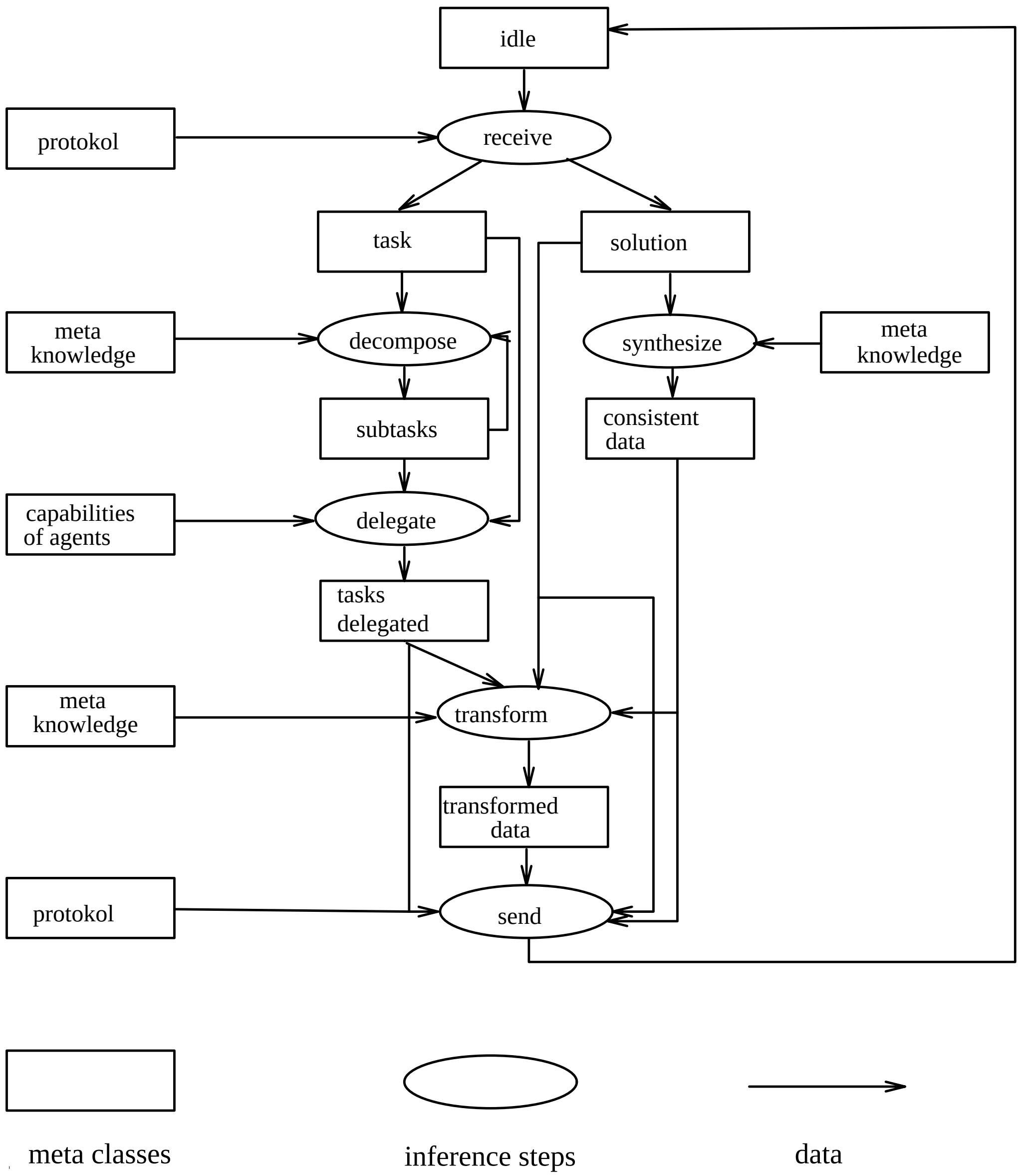

Figure 6: Inference Knowledge of the supervisor agent.

an addition of points to express the evidence of a diagnosis. The following categories are given:

category	total number	category	total number
P7	999	N7	-999
P6	80	N6	-80
P5	40	N5	-40
P4	20	N4	-20
P3	10	N3	-10
P2	5	N2	-5
P1	1	N1	-1
P0	0		

P7 means, that the diagnosis is true; N7, that it is excluded. Depending on the total number of points a diagnosis has, the evidence of a diagnosis can be expressed as follows:

t otal number of points				probability category
0	-	50	points	= neutral
50	-	60	points	= probable
60	-	100	points	= very probable
	>	100	points	= true

There is only a small interval of points between 50 and 100, were the evidence changes from neutral to true. In order to transform the representation of the evidence from EMYCIN to D3 we must find a mapping function with the same behaviour as the function in figure 7. A function to satisfy these conditions is a *tanh* function or a growing function with suitable parameters. The values over 500 are mapped to 1 and the values less than -500 to -1. A mapping from Emycin to D3 could be the inverse function with the results truncated to an integer number.

The rules to resolve inconsistent knowledge are also represented in the domain knowledge. Generalized annotated programms (GAP) [10, 11] provide a rich framework for reasoning with simple temporal informations and uncertainty and distributed assumption-based reason maintenance. Using this logic one can model the meta knowledge of a supervisory agent, eg. which agent should be prefered if two agents deliver contradictory opinions. Continuing our sample scenario, the following situation in a hospital is given: a patient with heart disease is brought to the hospital. At the reception the patient is asked about his private data and the history of his illness. But there exists also a local database in the heart disease ward with special data on the patient. Our patient has been in the hospital before; therefore there exist also old data for him. A diagnosis expert system asks for information about the history of his illness. The supervisor delegates this task to the local database and the reception database. If the local database is not updated, then there are maybe different results concerned with the history of the illness. This situation must be resolved. The supervisor chooses the most recent information.

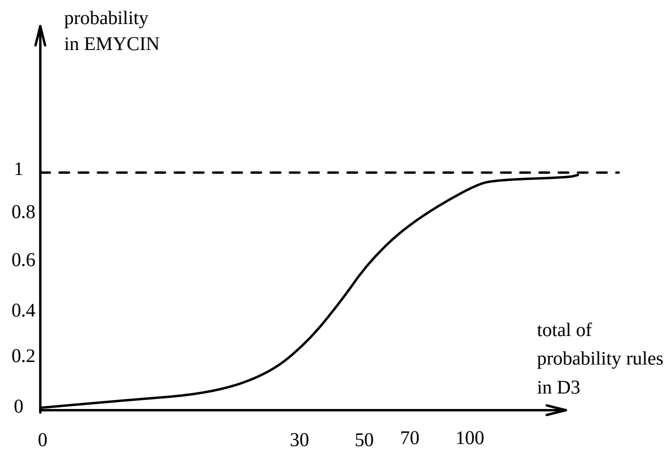

Figure 7: Behaviour of the mapping function.

In the following example a sympton *sym* is asked, which is represented in the database with a time stamp and a boolean value. This value shows if at the stamped time the symptom occured or not.
The rules in GAP are as follows (s represents the supervisor agent):

data base 1

$$\text{symptom(sym)} : 25.8.1993, \mathbf{t} \leftarrow .$$

data base 2

$$\text{symptom(sym)} : 12.12.1993, \mathbf{t} \leftarrow .$$

Solution of conflict:

$$\begin{aligned}\text{symptom(X)} : [\{s\}, T, Y] \leftarrow\ & \text{symptom(X)} : [\{D\}, T, Y]\ \& \\ & \text{symptom(X)} : [\{1,2\} - D, T', Z]\ \& \\ & T \geq T'.\end{aligned}$$

Another example might be the diagnosis of an infection. Two expert systems determine, if there is an evidence of an infection with bacteria and which type of bacteria it is. If these two expert systems give different evidences of the infection with a special type of bacteria, the most probable infection is chosen. But it is chosen only if there is no negation of the infection by the laboratory.

XPS_1 and XPS_2 deliver the following diagnosis of an infection with the bacterias A,B and C with evidences. The results of the analysis of the laboratory (lab) are also given.

XPS_1	XPS_2	laboratory
infection(A) : 0.5 ←.	infection(A) : -0.5←.	lab(A) : 0 ←.
infection(B) : 0.3 ←.	infection(B) : 0.8 ←.	lab(B) : -1 ←.
infection(C) : -0.5 ←.	infection(C) : 0 ←.	lab(C) : 0 ←.

Solution of conflict:

infection(Z) : $[\{s\}, 1] \leftarrow$ lab(Z) : $[\{3\}, 1]$.
infektion(Z) : $[\{s\}, X] \leftarrow$ infection(Z) : $[\{T\}, X]$ &
infection(Z) : $[\{1, 2\} - T, Y]$ & $X > Y$ &
not(infection(W) : $[\{V\}, R]$ & $R > X$ & not(lab(W) : -1)) &
not(lab(Z) : -1.).

With this rules we can conclude that the infection chosen is infection A.

3.6 Relationships between the models

Now, we have a look at the landmark states and the relationships between them. There are internal relationships, which are relationships within the models, and external relations, which are relationships between the models. In figure 8 there are the most important relationships between the landmark states. The development of the CommonKADS model set is determined by them. To construct the models they must be achieved in this sequence. Firstly, we must define the organization model and secondly the task decompostion in the task model. After fixing the functionality of a task, the agents must be determined in order to specify the ingredients. Based on the ingredients and agents, the communication model and finally the expertise model can be defined. This process is finished be the development of the design model, since decisions concerning the future CKBS shell, such as architectural issues, are not relate to the other models.

4 Related work

Smith and Davis [1, p. 61] described a framework for cooperation in distributed problem solving, especially cooperation by task sharing. The Contract Net is based on this ideas. They described three phases of distributed problem solving: the problem decomposition, the subproblem solution and the synthesis of answers.

As shown in our report this ideas are reused in the CommonKADS method and in the elaboration of the Expertise Model. The Contract Net bases on a loosely coupled collection of knowledge sources. A contract is an explicit agreement between a node that generates a task (the manager) an a node willing to execute the task (the contractor). The manager is responsible for monitoring the execution of a task and processing the results of its execution. The contractor is responsible for the actual execution of a task. Every node in the Contract Net can be a manager or a contractor depending on the actual task to solve.

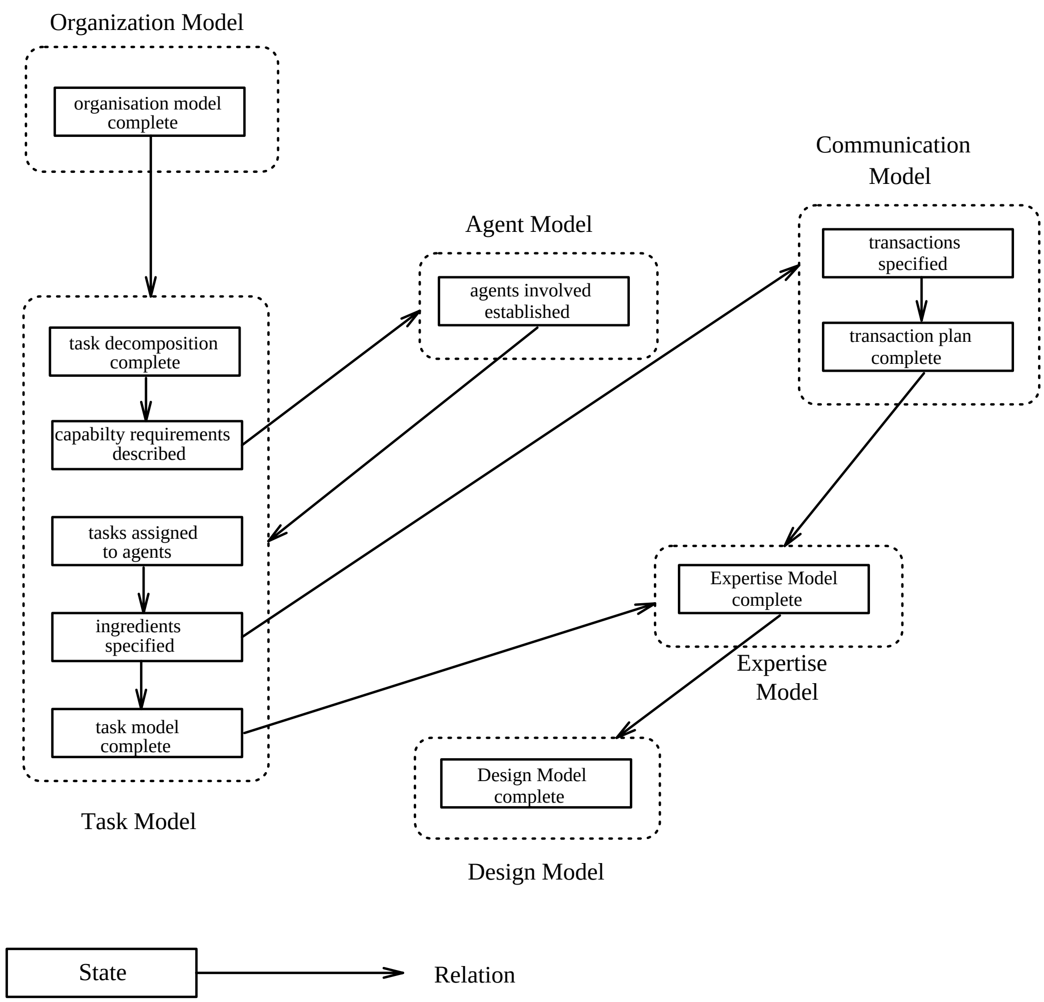

Figure 8: Relationships between and within the models

The CommonKADS method supports the knowledge acquisition for a Contract Net. Using the task decompostion and the capabilities agents have, contracts can be made and subtasks executed. The manager of a task is responsible for the synthesis of solutions described in the Expertise Model. But with the CommonKADS method the architecture need not to be a decentralized one. In a centralized architecture the central node performs the decomposition and the synthesis of the solutions. In CommonKADS the knowledge is described independently of the future architecture.

In the last KADS meeting an expansion of KADS to DAI scenes was introduced: SOM (strategic oriented model building) [4], which is a methodology to acquire the experience of experts and the problem to be resolved. The method is similair to the task decomposition in CommonKADS and the description of the attributes of the tasks in the CommonKADS mthodology. It gives hints on how to fill the templates and to acquire the domain knowledge for each task.

In SOM the problem is decomposed and every "step" in the problem is related to the other steps. Through that, a plan is defined on how to solve a problem. CommonKADS goes farther. With the landmark and transition states a general method to acquire knowledge is propagated. SOM could be integrated in the CommonKADS method to define the task model and the domain knowledge. The aspects of agents on the one hand, and the communication and cooperation between agents on the other hand, are not developed in SOM. There is only one single big model to describe the knowledge and no division in separate models as in CommonKADS. How to analyze and resolve the cooperation and communication problems is not part of SOM.

5 Conclusion and further research directions

In the former sections we have illustrated the feasibility of CKBS knowledge acquisition using CommonKADS. As our example show the rate of exactness of the models in CommonKADS [6] is well balanced, it's neither too detailled nor to rough. The templates are sufficient to fulfill the needs for CKBS knowledge acquisition. With minor extension we were able to apply the whole CommonKADS machinery in a macroscopic sense. CommonKADS supports the decomposition of a CKBS problem into subtasks and the identification of inter-agent cooperation and communication. The supervisor is responsible for the coordination of cooperation and communication and can be modelled independently from the future architecture of the CKBS which might be centralized or decentralized. In a centralized architecture the supervisor agent takes the role of the central node. In a decentralized architecture every node inherits the knowledge of the supervisor. With the supervisor agent described in our work, every agent might be a manager or a contractor as proposed in the Contract Net [1].

In a microscopic sense the CommonKADS methodology does not support issues such as the transformation of different modes of uncertain reasoning [14] and synthesis of different solutions from different agents. This synthesis can be formalized using generalized annotated logic [10], enabling temporal and inconsistent reasoning. The transformation functions and the GAPs (generalized annotated logic programms) are used in the domain knowledge of the supervisor agent's expertise model. The inference and task knowledge of the supervisor can be reused for other CKBS problems. Part of an on-going research in our group is the implementation of a CKBS shell based upon the above mentioned GAP framework [3].

References

[1] Bond, A. (Ed.): *Readings in Distributed Artificial Intelligence.* Morgan Kaufman Publishers, 1991.

[2] Deen, S.M.: *Cooperating Agents - A Database Perspective.* Proceedings of the International Working Conference on Cooperating Knowledge Based Systems CKBS '90, S.1-29, 1990.

[3] Calmet, J.; Messing, B.; Schü, J.: *A Novel Approach towards an Integration of Multiple Knowledge Sources.* International Symposium on the Management of Industrial and Corperate Knowledge ISMICK 1993.

[4] Dorfstecher,G.: *Strategieorientierte Modellbildung: SOM.* Bericht zum KADS Workshop 3, 1993.

[5] de Greef, P.; Breuker, J.; de Jong, T.: *Modality, an Analysis of Functions, User Control and Communication in Knowledge Based Systems.* ESPRIT Projekt P1098 KADS Ident UvA-A4-PR-004, 1988.

[6] de Hoog,P.; Martil, R. et al.: *The CommonKADS Model Set.* Esprit Projekt P5248 KADS-II, Ident: Kads-II/WP I-II/RR/UvA/018/4.0, 1992.

[7] Porter, D.: *Overview of the Differences between KADS and CommonKADS.* ESPRIT Projekt P5248 KADS-II, Ident: KADS II/T5.1.2/PP/TRMC/002/0.1, Juni 1992

[8] Puppe, F.: *Einführung in Expertensysteme.* Springer Verlag 1988.

[9] Schupeta, A.: *COSMA, ein verteilter Terminplaner.* Beiträge zum Gründungsworkshop der Fachgruppe VKI. Saarbrücken 1993. Hrsg. J. Müller.

[10] Subrahmanian, V.S.: *Amalgamating Knowledge Bases.* Interner Bericht des Department of Computer Science, University of Maryland, 1993.

[11] Subrahmanian, V.S. *On the Semantics of Quantitive Logic Programms.* Proc. 1987 IEEE Symp. on Logic Programming, pps 173-182.

[12] Weih, H.-P.: *Modellierung eines Beratungssystems mit CommonKADS und Untersuchung der Verwendung von CommonKADS für verteilte Künstliche Intelligenz.* Diplomarbeit Universität Karlsruhe 1994.

[13] Wielinga, B.; van de Welde, K. et al.: *Towards a Unification of Knowledge Modelling Approach.* ESPRIT Projekt P5248 KADS- II, Ident: KADS-II/T1.1/UvA/RR/004/4.0, 1993.

[14] Zhang, C.: *HECODES.* Thesis for the Degree of Doctor of Philosophy of the University of New England, 12.1990.