Bridging Gaps in Models of Expertise

K. Christoph Ranze and Heiner Stuckenschmidt

Center for Computing Technologies, Bremen University,
P.O.B. 330440, D-28334 Bremen, Germany
{kcr,heiner}@tzi.de
http://www.tzi.de

Abstract. This paper presents an approach to the explicit integration of uncertain reasoning mechanisms into conventional KADS-based models of expertise. We assume that uncertainty is present in the model of expertise in the sense that terminological knowledge is available but some assertions cannot be determined due to uncertainty. These gaps can be bridged in a three-step inference scheme using a general model of uncertainty in which knowledge is represented as sets of hypotheses.
Our work is based on Shenoy's valuation-based systems and on Pearl's structural equation model of causality. Combining these two approaches we obtain a model of uncertainty including a scheme for uncertain inferences. The resulting structure can be used to integrate the specification of uncertainty into conventional models of expertise.
The inference scheme presented in this paper can be seen as a first step towards the implementation of uncertainty reasoning specified in a model of expertise using graph-based approaches for reasoning under uncertainty like Bayesian networks.

1 Introduction

While models of expertise form the essential part in model-based knowledge engineering, most of them lack explicit concepts for the specification of uncertainty. In real world applications adjectives like 'probable, 'possible' or 'incomplete' are attached to domain knowledge and data. We summarize these phenomena of non-categorical knowledge as *uncertainty*. Investigating different KADS-based knowlege engineering approaches we found no sophisticated formalism for explicit representation of different phenomena of uncertainty. On the other hand there is a huge amount of elaborated (numerical) calculi representing and processing uncertain knowledge in application systems.

From this point of view the aim of the presented work is twofold. On one hand the explicit integration of uncertain knowledge in model-based knowledge engineering frameworks is an important topic. On the other hand we try to elude gaps within ordinary models of expertise based on the well-known KADS approach [26]. Those gaps usually originate from different kinds of uncertainty. As the major focus of this paper we define an uncertain inference scheme filling these gaps by determining assertions from sets of hypotheses.

The definition of a theoretical basis for an overall model of uncertain expertise is a very challenging task due to the conceptual differences between existing models of expertise and the leading approaches in uncertain reasoning. While most models of expertise are based on a logic-based semantics ([5] offers a brought survey), causality has proven to be one of the most promising approaches to describe uncertain knowledge represented by sets of hypotheses. Probably the most cited reference on this topic is [11]. We have chosen to take the best of both worlds and use an existing expertise model and a separated causal structure to represent uncertainty in this model. The integration of these different foundations is done by a mapping which can be seen as a translation between the different semantics in use.

As a base we define a general model of uncertainty based on the terminological knowledge specified in an ordinary model of expertise. In case of uncertainty some assertions are missing. The corresponding knowledge instead is represented as a set of possible hypotheses. In the model of uncertainty elements from the model of expertise are then "evaluated" with respect to their uncertainty in a process of uncertain reasoning about valuations of hypotheses. These evaluations serve as a basis for the determination of assertional knowledge filling the gaps in the ordinary model of expertise. In order to preserve the structure of the model of expertise, a distinction is made between domain knowledge, inference knowledge, and task knowledge in the model of uncertainty as well. The resulting structure can be used to integrate the specification of uncertainty into a model of expertise.

2 A Framework for Modelling Uncertain Expertise

In this section we first present a generalized notation of KADS-like models of expertise. Secondly we review a framework for the specification of uncertain expertise proposed in [14], which is based on two existing approaches. Shenoy's valuation-based systems are used to to evaluate possible assertions represented as sets of hypotheses whereas Pearl's structural equation model of causality is used to impose a structure on the valuated assertions. After a short presentation of these approaches we show their combination leading to our causal model of uncertainty, that captures uncertain expertise in terms of valuation functions and structure equations.

2.1 KBS Specification Languages

The starting point for our work was the intention to describe the reasoning process of an expert in terms of a conceptual model of expertise. This aim coincides with the rationals of model-based knowledge engineering. Therefore it is not surprising that researchers in the field of knowledge engineering have developed many different languages for describing such models that are well-suited to serve as a basis for the specification of uncertain reasoning, though none of these languages is able to handle uncertainty in the way we claim to be adequate for our needs.

In the following we give a rough overview over existing specification languages for knowledge-based systems that is mainly based on two survey articles by van Harmelen and others [23] [5] and incorporates some recent approaches, that are not covered therein. Afterwards we take a closer look at the structure of languages based on the KADS model of expertise which we adopt for our approach. In the last paragraph of this section we present the language $(ML)^2$ we have chosen as an example language for applying our approach. In particular we show how $(ML)^2$ instantiates the scheme of the KADS-model of expertise.

Overview of Existing Languages The nature of a specification language strongly depends on the purpose it has been developed for. We roughly distinguish three groups of existing languages. The first group consists of general purpose languages which also includes languages for multi-agent systems, since in our view every knowledge-based system can be seen as an intelligent agent. The second group which will be of major interest in our further investigation contains languages developed to describe models of expertise well-known from leading approaches to knowledge engineering. In the last years ontologies and problem-solving methods have become the major focus of knowledge engineering approaches. While most languages for models of expertise are unsuited especially for specifying the dynamics of problem-solving methods, some new languages have been developed that concentrate on this special topic forming a third group of specification languages. In the literature the following languages are mentioned:

general purpose: DESIRE, MC, MILORD II, QIL [23]

models of expertise: FORKADS, KARL, $K_{bs}SF$, $(ML)^2$, MODEL-K, MoMo, OMOS [5]

problem-solving methods: MCL(MLCM, MLPM) [7], NewKARL [2], OCML [10], UPML [6]

These languages also differ in modelling ability and formal strength. While so-called conceptual languages offer the flexibility and expressiveness of restricted natural languages, formal languages make a trade-off between the two dimensions using formal semantics without making contributions to operationalisation. The modelling ability of operational languages is restricted to allow for execution of the specification. These differences can be exploited to support the different stages of the knowledge engineering process. While conceptual languages can be used for acquisition and structuring of knowledge, formal languages are well-suited for the specification of the aspired system while operational languages can be used to build a prototypical system.

For the scope of this paper we restrict ourselves to formal languages that aim at the formalisation of KADS-like models of expertise.

The KADS Model of Expertise To enable translations between the model of uncertainty and the model of expertise we first have to define the language used to describe expertise knowledge. The definition of this language is provided by the terminology of the model of expertise.

Models of expertise are typically subdivided into three kinds of knowledge: domain, inference, and task knowledge as defined in the KADS model of expertise [25]. In the following we define an abstract terminology for models of expertise, which will serve as a basis for our model of uncertainty in the later chapters.

Those parts of the real world relevant to the given task are described with their properties within the domain model. The formal specification of this knowledge is realized by a set of ontological primitives enabling the user to define complex structures.

Definition: 1 (Domain Model) *Let $\mathcal{ABB}(A, B)$ be the set of all mappings from A to B. A domain model $\mathcal{DOM} = (\mathcal{K}, \mathcal{I}, \mathcal{W}, \mathcal{A}ttr, \mathcal{R})$ is defined in terms of*

(i) a set $\mathcal{K}$ of concepts and a set $\mathcal{I}$ of instances of these concepts

(ii) a value set $\mathcal{W}$

(iii) a set $\mathcal{A}ttr \subseteq \mathcal{ABB}(\mathcal{I}, \mathcal{W})$ of attributes,

(iv) a set $\mathcal{R} \subseteq \mathcal{I}^n \cup \mathcal{K}^n$ of relations

Based on the modelled elements there are elementary operations performing single steps of the problem-solving process. These operations called *knowledge sources* are assumed to be primitive and fully determined by an input-/output specification. A knowledge source operates on elements from the domain layer. These elements are described through meta-classes (input-, output- and static role) which are placeholders determining the role the element plays in the problem-solving process and the type of domain objects that can play this role.

Definition: 2 (Inference Model) *Let $\mathcal{ABB}(A, B)$ be the set of all mappings from A to B. An inference model $\mathcal{INF} = (\mathcal{IR}, \mathcal{SR}, \mathcal{OR}, \mathcal{KS})$ is defined as follows*

(i) a set $\mathcal{IR}$ of input roles and a set $\mathcal{OR}$ of output roles

(ii) a set $\mathcal{SR}$ of static roles

(iii) a set $\mathcal{KS} \subseteq \mathcal{ABB}(2^{\mathcal{IR}} \times 2^{\mathcal{SR}}, \mathcal{OR})$ of knowledge sources

The task layer contains knowledge about how the elementary inference steps can be combined and executed to achieve a certain goal. This knowledge is organized in tasks which are compositions of subtasks including control knowledge about their execution in order to achieve the goal of the main task. Primitive tasks which do not have subtasks show a one-to-one correspondence to knowledge sources within the inference layer.

Definition: 3 (Task Model) *A task model $\mathcal{TASK} = (\mathcal{T}, Subtask, \mathcal{E}, goal)$ is defined as follows.*

(i) $\mathcal{T}$ *is a set of tasks and* $\mathcal{E}$ *is a set of task results*

(ii) $Subtask \subseteq \mathcal{T} \times \mathcal{T}$ *is a subtask relation*

(iii) $goal : \mathcal{T} \to \mathcal{E}$ *is a mapping assigning results to tasks*

Together these three layers form a model of expertise that claims to capture all aspects of expert reasoning relevant to the development of knowledge-based systems. A common model is achieved by connecting the different layers in the sense that the roles of inferences are filled with domain knowledge and tasks are executed by applying inferences which produce a result corrsponding to the task's goal. Based on the terminology, concrete assertions about the reasoning process are specified in terms of axioms over the language used. Beside first-order logic, meta-logic and dynamic- or modal logic are possible formalisms used to describe expertise knowledge (again see [5]).

$(ML)^2$: An Example Language To illustrate our approach we have chosen the logic-based specification language $(ML)^2$ [22] which has been developed in the KADS project to formalise conceptual models of expertise. This paragraph gives a brief overview of the concepts used in $(ML)^2$ to describe the different elements of a model of expertise mentioned in the preceeding section.

In $(ML)^2$ order-sorted logic is used to describe the domain layer. Concepts are represented by sorts imposing a typing on constants which stand for instances of those concepts as well as for functions which describe attributes of instances and predicates which are interpreted as relations between the instances. The domain model is divided into subtheories connected by import relations.

The inference layer is realised using a meta-logic over the domain. All kinds of roles are described by meta-level terms derived from the domain by non-standard naming relations [24]. Knowledge sources are meta-level predicates over these terms. Each inference predicate is specified in a meta-level theory importing the definitions of the role terms.

Quantified dynamic logic is used to model the task layer. Each task corresponds to a program expression in this logic. The execution of subprograms imposes a subtask relation on the different programs. The goal assignment is realized by special predicates of the dynamic logic.

2.2 Valuation-Based Systems

A valuation-based system is a kind of a knowledge-based system in which knowledge is represented by variables and special valuation functions [16]. A valuation attaches a degree of certainty to configurations of variables. Degrees of certainty are often represented by numbers from the interval $[0,1]$. Variables of a valuation-based system are denoted by small letters, the valuation functions with corresponding capitals (therefore G is the valuation for the variable g). If g is a variable then W_g represents the set of all possible values for g. Relations such as rules or causal relations between variables are represented by valuations connecting more than one variable, whereas propositions are modelled by valuations for single variables.

In a valuation-based system, inference is performed by two operations, combination and marginalization. One can describe a valuation-based system defining these two operations and a set of valuations, modelling the knowledge in use.

Definition: 4 (Valuation-Based System) *Let V_i be valuations of variables v_i, and Ψ_i sets of truth-values. A valuation-based system is a 3-tupel*

$$(\{V_1, \cdots, V_m\}, \otimes, \downarrow)$$

where $\{V_1, \cdots, V_m\}$ is a set of valuations of the form $V_i : W_{v_i} \to \Psi_i$, $\otimes$ is a combination operator and $\downarrow$ is a marginalization operator.

The combination of two valuations corresponds to logical conjunction and is therefore used to aggregate knowledge. The application of the combination operator combines two valuations H and G and creates a new valuation $H \otimes G$ which is a valuation for the union of the corresponding variables g and h if this is permitted by the nature of the combined valuations. Marginalization corresponds to logical disjunction and stands for the coarsening of knowledge. Marginalization is used to calculate the valuation of a subset of variables $h - \{x\}$ from the valuation for the whole set h of variables.

2.3 Pearl's Model of Causality

In the structure equation model proposed by Pearl [13] the values of involved variables are seen as the result of a function whose arguments are the values of other variables in the model. Variables are separated into internal variables V_i and external variables U_i. The values of internal variables may be determined in principle, whereas the values of the external variables are completely hidden and only considered in terms of a distribution of certainties. The value of an internal variable is therefore given by the equation $V_i = f(\mathbf{pa}_i, \epsilon)$, where $\mathbf{pa}_i$ is a set of internal variables which influence V_i and ϵ describes a degree of uncertainty induced by external variables. The equations for all internal variables establish a causal model which is completely described by these equations.

Definition: 5 (Causal Theory) *A causal theory is a 4-tuple*

$$T = \langle V, U, D(u), \{f_i\} \rangle$$

with

(i) $V = \{V_1, \cdots, V_n\}$ is a set of internal variables which can be observed

(ii) $U = \{U_1, \cdots, U_m\}$ is a set of external not definable variables inducing uncertainty

(iii) $D(u)$ *is a distribution of valuations over* U

(iv) $\{f_i\}$ *is a set of structural equations of the form*

$$V_i = f_i(\mathbf{pa}_i, \epsilon), i = 1, \cdots, n \tag{1}$$

with $\mathbf{pa}_i \subseteq V, V_i \notin \mathbf{pa}_i$

This definition guarantees the intuitive characteristics of causality. Temporal ordering and the definition of relevant contexts are done by the use of variables as parameters of the function defining the value of another variable, which demands the variables in $\mathbf{pa}_i$ to be defined before V_i. They also describe the relevant context for V_i.

2.4 Modelling Uncertain Knowledge

Using these two approaches we are able to represent uncertain expertise. The essential descriptive elements are pairs with sets of hypotheses which are represented as value sets W_{v_i} of variables v_i and valuation functions V over these value sets. In the following these pairs will be referred to as "phenomena of uncertainty".

Definition: 6 (Phenomenon of Uncertainty) *A phenomenon of uncertainty UP is a pair*

$$UP = (W_v, V)$$

where W_v *is a set of hypotheses and* V *is a valuation function on this set.*

As a rule, on the basis of observations only a small number of the sets of hypotheses may be evaluated directly. Therefore, a mechanism for the propagation of the given valuations is essential. Especially for this purpose causal schemata are appropriate since they elegantly describe dependencies, thus leading to an efficient calculation. In our approach the functions of the structural equation model are instantiated with concrete operations which may serve for the modelling of domain knowledge.

Definition: 7 (Operator) *Let* $\mathcal{UP}$ *be a set of phenomena of uncertainty, an operation on* $\mathcal{UP}$ *is given through the mapping*

$$OPS : \mathcal{UP}^n \times \Psi \rightarrow \mathcal{UP}$$

If $p_1, \cdots, p_n$ *are phenomena of uncertainty, the outcome* r_{OPS} *of the operation is defined as follows*

$$r_{OPS} = OPS(p_1, \cdots, p_n, \psi)$$

$\mathcal{OPS}$ *denotes the set of all operations,* $\mathcal{RES}$ *denotes the set of all outcomes.* ψ *denotes the degree of uncertainty attached to the causal relation. Usually* $\mathcal{OPS} = \{\wedge, \vee, \neg\}$ *is defined as a default.*

As a basis the classical logic operators are an obvious choice, since they offer manifold modelling possibilities and their significance has already been investigated [3].

Actions may also be represented by means of the causal scheme, i.e. for the modelling of inferences onto phenomena of uncertainty. In this case input and output roles are expressed by causes and effects of the causal connection. These inferences will generate knowledge that hitherto has not been available in the domain model. In the model of uncertainty this new knowledge will also be specified by a phenomenon of uncertainty. Once the (uncertain) result of an inference has been defined in terms of phenomena of uncertainty, an inference within the model of uncertainty may again be modelled as a function mapping a set of phenomena of uncertainty corresponding to the input and static roles of the inference on a further phenomenon of uncertainty.

Definition: 8 (Uncertain Knowledge Source) *Let $\mathcal{UP}$ be a set of phenomena of uncertainty, an uncertain knowledge source is defined as a mapping*

$$UKS : \mathcal{UP}^n \times \Psi \rightarrow \mathcal{UP}$$

If $p_1, \cdots, p_n$ are phenomena of uncertainty and ϵ as mentioned above, then

$$e_{UKS} = UKS(p_1, \cdots, p_n, \psi)$$

denotes the result of UKS. $\mathcal{UKS}$ denotes the set of all uncertain knowledge sources, $\mathcal{ERG}$ the set of all results.

In the model of uncertainty, uncertain knowledge sources play a similar role as knowledge sources in a conventional model of expertise, since they are transferring the examined knowledge to a different level. The results do not reflect direct or indirect observations any more, but rather the results of a problem-solving process which, independent of the environment, takes place at the actor's.

2.5 Combining the Models

Figure 1 shows the architecture used to describe uncertain expertise, which results from the scheme described above. It consists of two separate models connected through mappings which serve as interfaces for knowledge that is used in both models. In the following the two mappings are briefly described on an informal level. A more detailed investigation of the connection can be found elsewhere ([19]).

Reference Mapping The reference mapping is defined as a transformation from the model of uncertainty into the terminology of the model of expertise, which has been chosen for the description of the reasoning process. The major function of this mapping is to relate phenomena of uncertainty to modelling primitives used in the model of expertise. The reference mapping is split up into

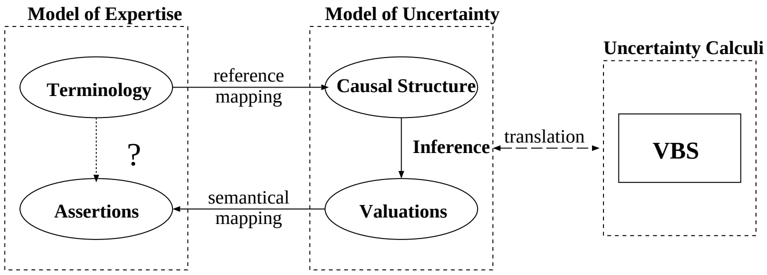

Fig. 1. Architecture of the Model

three submappings, each of them establishing the connection to one of the three layers of a KADS-based model of expertise [26]. The mapping $ref[task]$ maps phenomena onto results of the task layer while the mapping $ref[inf]$ relates phenomena to roles in the inference layer. The connection to the domain layer is established by the mapping $ref[dom]$. To be precise, all phenomena own a connection to the domain layer because phenomena of uncertainty always represent domain knowledge. An additional connection to one of the other layers exists only if this domain knowledge is used in the reasoning process.

As every phenomenon owns a connection to the domain layer, this connection can be used to determine the role of a phenomenon in the problem-solving process by attaching to each phenomenon an entity and a function from the domain model. The entity referred to as $ref[obj]$ can either be a concept or an instance of a concept. The referred function can be an attribute or a Boolean function over elements of the domain. The latter is mainly used to relate phenomena to relations over concepts or instances. Additionally, the role of causal relations between phenomena is also made clear by the reference mapping. For this purpose, causal connections are either mapped onto operators defined on domain knowledge (e.g. Boolean operators) or on tasks and inferences, respectively. In the former case, the causal relation is referred to as operator, in the latter case we use the term *uncertain knowledge source.*

Semantical Mapping So far all we know about the phenomena of uncertainty is the entity-function pair they refer to. What needs to be done is to form an assertion about knowledge concerning the reasoning process using the language provided by the application of the referred function to the referred entity. As these components are known only the result has to be fixed. First of all it is not possible to determine this result exactly due to the uncertainty that has to be captured by the model of uncertainty. So instead of a single result, each phenomenon provides a set of possible results – the set of hypotheses. As long as none of the hypotheses has been selected as being true, the semantics of a phenomenon are given by the disjunction of assertions, each defining one of the hypotheses to be the result of the application of the referred function to the referred entity.

While the concrete appearance of the translation of a phenomenon depends on the terminology used, some general predications about the integration of the translation into the model of expertise can be made. First of all, the translations form axioms over the terminology referred to by the translated phenomenon. Therefore the translation, which can be seen as the semantics of the phenomenon with respect to the model of expertise, is located in the set of axioms of the theory also inclosing the terminology used. This theory is called reference theory of the translated phenomenon and represents the interface between certain and uncertain expertise knowledge.

3 Bridging Gaps with Uncertain Inferences

To use the model of uncertainty for the specification of uncertainty in a problem-solving process, the process of reasoning has to be described formally. This can be done using an existing specification language. We assume that uncertainty is present in the model of expertise in the sense, that terminological knowledge is available but some assertions cannot be determined due to uncertainty. These gaps in the model can be bridged in a three-step approach using our model of uncertainty.

Determination of the language to be used in the model of uncertainty is the first step. For this purpose phenomena of uncertainty and uncertain knowledge sources are explicitely connected to terminological constructs in the model of expertise. The connection is done via a reference mapping from the model of uncertainty into the terminology of the expertise model. The referred concepts provide a language to describe the meaning of phenomena and uncertain knowledge sources.

Uncertain inference is the second step in which inference is carried out within the model of expertise deriving valuations of previously unknown phenomena. There is a wide range of calculi which can be used for this purpose, because inference in this model is equivalent with inference in valuation-based systems. Therefore existing approaches for describing well-known calculi of uncertainty in terms of a valuation-based system can be used to derive valuations.

Determination of assertional knowledge is the last step that uses the results of uncertain inferences to state axioms about the knowledge to be used in the problem-solving process. This step depends on a semantical mapping determining the meaning of elements in the model of uncertainty in relation to the model of expertise. Once a hypothesis is chosen to be accepted as reflecting most appropriately the true state of the world, the semantical mapping determines the related axiom that can be used in the model of expertise to perform further reasoning.

Both the determination of the language and of the assertional knowledge have been sketched via mappings between the models in section 2.5. Their detailed description is beyond the scope of this paper. We now have to focus on the definition of an inference scheme determining assertions from sets of hypotheses. For this we first have to define how inference is performed within the model of uncertainty. In second place we have to state how knowledge is transferred from the model of expertise to the model of uncertainty and vice versa.

3.1 Foundations

The basic idea behind our approach is to use the inference scheme offered by valuation-based systems since it offers manifold possibilities to map well-known numerical calculi. As an example we use triangular norms to instatiate the defined operations.

Inferences in VBS In valuation-based systems as described in 2.2, inference is performed by two operations, combination and marginalization. The combination of two valuations corresponds to logical conjunction and is therefore used to aggregate knowledge. The application of the combination operator combines two valuations H and G and creates a new valuation $H \otimes G$ which is a valuation for the union of the corresponding variables.

Definition: 9 (Combination) *Let v be the set of all valuations, then the combination operator is a mapping $\otimes : v \times v \to v$ with*

(i) *if G and H are valuations for g and h then $G \otimes H$ is a valuation for $g \cup h$*

(ii) *if either G or H is not a correct valuation, then $G \otimes H$ is also not a correct valuation*

(iii) *if G and H are correct valuations, then $G \otimes H$ may or may not be a correct valuation, depending on the nature of the valuations*

Marginalization corresponds to logical disjunction and stands for the coarsening of knowledge. Marginalization is used to calculate the valuation of a subset of variables $h - \{x\}$ from the valuation for the whole set h of variables.

Definition: 10 (Marginalization) *For each set of variables h there exists a mapping $\downarrow (h - \{x\}) : v_h \to v_{h-\{x\}}$ with*

(i) *if H is a valuation for h, then $H^{\downarrow(h-\{x\})}$ is a valuation for $h - \{x\}$*

(ii) *$H^{\downarrow(h-\{x\})}$ is a correct valuation if and only if H is a correct valuation.*

$H^{\downarrow(h-\{x\})}$ is called the marginal for $h - \{x\}$.

Using these two operators, a large spectrum of inference mechanisms can be simulated in a common framework [17]. It is assumed that the valuation of a single variable has to be calculated from other valuations in a system of connected variables. This is performed by combining the known valuations of the system and calculating the marginal for the variable whose value has to be determined. Most of the existing calculi for handling uncertainty fit into the framework of valuation-based systems. This has been done by Shenoy and Shafer [18] for Bayesian probabilities and by Dubois and Prade [4] for possibility theory.

Triangular Norms Triangular norms [3] provide axiomatic descriptions of functions which are convenient for the definition and execution of operations on uncertain knowledge. They serve as generalizations of the classical logical operators conjunction, disjunction, and negation. Unlike these classical operators triangular norms state functions of the form $f : [0,1] \times [0,1] \rightarrow [0,1]$. The first group of these triangular norms are t-norms, which serve as a generalization of logical conjunction.

Definition: 11 (T-Norm) *A two-placed operator* $T : [0,1] \times [0,1] \rightarrow [0,1]$ *is denoted as t-norm, if the following conditions hold*

T1: $T(0,0) = 0$

T2: $T(a,1) = a = T(1,a)$

T3: $a \leq c \wedge b \leq d \Rightarrow T(a,b) \leq T(c,d)$

T4: $T(a,b) = T(b,a)$

T5: $T(a,T(b,c)) = T(T(a,b),c)$

These axioms assure the properties of conjunctions with respect to the limits '0' and '1'. Functions with these properties are adequate for the aggregation of premises as well as for sequential combination. Due to their associative nature t-norms could be applied to more than two variables.

$$T(x_1, \cdots, x_n) = T(T(x_1, \cdots, x_{n-1}), x_n) \tag{2}$$

Analogously t-conorms serve as axiomatic descriptions of functions describing generalizations of logical disjunction. T-conorms utilize a parallel combination of valuations.

Definition: 12 (T-Conorm) *A two-placed operator* S *is denoted as t-norm, if the following conditions hold*

S1: $S(1,1) = 1$

S2: $S(0,a) = a = S(a,0)$

S3: $a \leq c \wedge b \leq d \Rightarrow S(a,b) \leq S(c,d)$

S4: $S(a,b) = S(b,a)$

S5: $S(a,S(b,c)) = S(S(a,b),c)$

T-conorms preserve the properties of logical disjunction and could be applied to more than two variables likewise.

$$S(x_1, \cdots, x_n) = S(S(x_1, \cdots, x_{n-1}), x_n) \tag{3}$$

Here we abandon an axiomatic definition of the negation operator which is normally defined as $N(x) = 1 - x$.

As triangular norms are generalizations of the logical operators conjunction, disjunction, and negation, they could serve as operators for combination and marginalization within valuation-based systems.

On one hand the combination of single valuations is a direct application of a t-norm since those combinations are aggregations of premises. Therefore no further investigation of this operation is necessary. On the other hand, conclusions of rule-based statements are drawn with *modus ponens*. With triangular norms this is done via a sequential combination of a valuation of a single variable and a valuation over a set of variables. The latter represents the rule. At last, marginalization is performed by a t-conorm as a parallel combination of all configurations.

3.2 Uncertain Inference

Considering section 2 all we know so far about phenomena of uncertainty is their static representation. To gather new knowledge we have to define an 'uncertain' inference which mainly consists of the determination and execution of a valuation function for a given variable within the model of uncertainty. The requested function results from the causal relations and their valuations within the model. Starting with some certain phenomena, the phenomena of uncertainty and the contents of those phenomena of uncertainty, which serve as roles for inferences, are determined by means of operations. Thereupon the results of the uncertain knowledge sources are determined, which serve as elementary tasks. Repeating the processing of uncertain knowledge sources hierarchically, the result of the overall problem will be achieved.

Valuations The valuation of uncertainty in our model is represented by two different valuation functions. Both can be seen as valuation function in the sense of a valuation-based system. On one hand, the *valuation of hypotheses* attaches a degree of certainty to every element of a set of hypotheses. We use degrees of certainty from the interval [0..1] since they allow the definition of operators based on the basic arithmetical operations. The valuation functions are denoted as *cert* (certainty), so that $cert(x)$ denotes the valuation of element x of a set of hypotheses. To avoid confusion, we use X to denote the corresponding variable. Therefore $cert(X = x)$ denotes the valuation for the instatiation of X with x.

Definition: 13 (Valuation of hypotheses) *If X is a phenomenon of uncertainty with $W_X = \{x_1, \cdots, x_n\}$, then the valuation of the set of hypotheses of X results from the valuation function cert*

$$cert(X = x_i) \in [0, 1], \forall x_i \in W_X \tag{4}$$

On the other hand the *valuation of causal influences* valuates the strength of a causal relation between a specific value of a variable and a configuration of related variables. This valuation function is therefore based both on the value y of the goal variable Y and on the values $x_1, ..., x_n$ of the variables $X_1, ..., X_n$ from the set $\mathbf{pa}_Y$ of relevant variables. Valuations of causal influence are denoted as *cert* as well. The additional parameters are represented through the implication, whose conclusion is the goal variable.

Definition: 14 (Valuation of causal influence) *If $Y = f_Y(pa_Y, \epsilon)$ is a phenomenon of uncertainty and $X = pa_Y$, then the valuation of the causal influence from X to Y is given through the valuation function cert*

$$cert(f_Y(x_{i_1}, \cdots, x_{i_n}) \rightarrow Y = y_k) \in [0,1], \forall x_{i_j} \in W_{X_i}, y_j \in W_Y \tag{5}$$

An Inference Scheme The restriction of the truth values of the valuation functions to the interval [0..1] allows for the use of triangular norms as inference operations. In the following we sketch our approach to binding triangular norms with the valuations described above. The calculation of an element of a given set of hypotheses is performed in three steps, each one using one of the diffent kinds of operations on uncertain knowledge described above.

1. The first step is the aggregation of valuations of relevant variables to achieve the total valuation of a variables configuration. The realisation of the aggregation by use of triangular norms depends on the function used in the structural equation. In most cases the aggregation depicts a combination and is therefore represented as a t-norm. The parameters consist of valuations of hypotheses, each one an element of a set of hypotheses of a relevant variable. The results of an aggregation of a configuration $\{x_{j_1}, \cdots, x_{j_n}\}$ applying f_Y are denoted as $cert(f_Y(x_{j_1}, \cdots, x_{j_n}))$.

   Definition: 15 (Aggregation Operators) *Let the set of all causality functions within the model be given through $\mathcal{OPS} \cup \mathcal{UKS}$, whereas $\mathcal{OPS} = \{\wedge, \vee\}$. Let π^k denote the projection from a valuation vector of the parameter k within the causality function. Let T be a function with t-norm qualitities and S a function with t-conorm qualities. Hence an aggregation operator is defined through:*

   $$cert(f_Y(x_{j_1}, \cdots, x_{j_n})) = \begin{cases} T(\pi^1_{j_1}, \cdots, \pi^n_{j_n}) & if \quad f_Y \in \mathcal{UKS} \cup \{\wedge\} \\ S(\pi^1_{j_1}, \cdots, \pi^n_{j_n}) & if \quad f_Y = \vee \end{cases}$$

2. Achieving the result of the valuation of a configuration through aggregation of the valuations of hypotheses, one is able to calculate the causal influence of this configuration on a valuation of a hypothesis of the goal variable through sequential combination.

Definition: 16 (Sequential Combination) *Let $Y = f_Y(pa_Y, \epsilon)$ be a phenomenon of uncertainty and $X = pa_Y$ and T a t-norm, then a conditional valuation of Y is calculated through sequential combination as follows*

$$cert(Y = y_j | x_{i_1}, \cdots, x_{i_n}) \geq T(cert(f_Y(x_{i_1}, \cdots, x_{i_n}), m_{ij})$$

3. To achieve the final valuation of a hypothesis of the goal variable, one has to combine the causal influences of all configurations using parallel combination. This parallel combination is an application of a t-conorm to the resulting valuations of all causal influences calculated before. This step is necessary due to the fact that it is impossible to determine at the outset which configuration of the relevant variables is given. This step corresponds to a marginalization of the valuation function of the goal variable.

Definition: 17 (Parallel Combination) *Let $Y = f_Y(pa_Y, \epsilon)$ be a phenomenon of uncertainty and $X = pa_Y$ and S a t-conorm, then a valuation of Y is calculated through parallel combination as follows*

$$cert(Y = y_j) = S(\{cert(Y = y_j | x_{i_1}, \cdots, x_{i_n}) | (x_{i_1}, \cdots, x_{i_n}) \in W_{X_1} \times \cdots \times W_{X_n}\})$$

3.3 An Example Calculation

At this place, we would like to give a small example of how the inference scheme can be applied to determine an assertion that is needed to proceed in a problem-solving process. The example is a small cut-out taken from the SISYPHUS-I room assignment task. In an early stage of the problem-solving process one has to determine the order in which people in different positions are assigned to available rooms. This is quite easy provided there is a strict hierarchical relation between the different positions. We assume that this is not always the case, because different positions can obtain a high degree of informal importance for a research group without being at the top of the hierarchy. This informal importance leads to uncertainty in the knowledge base and makes it impossible to derive the most important position that has to be assigned before all others using a purely deductive inference. In our example, we want to decide whether the head or the manager of the group is the most important position concerning room assignment. The corresponding, admittedly very small model of uncertainty is shown in figure 2.

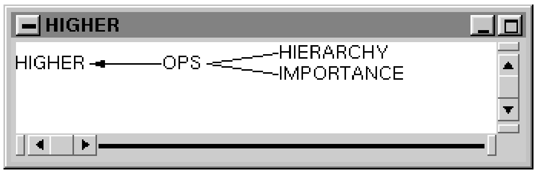

Fig. 2. Model structure used for the example

In our case the model of uncertainty only consists of two phenomena representing the hierarchy of the group and the degree of informal importance of the different positions respectively. These phenonema are connected by an conjunction operation to produce the phenomenon 'HIGHER' that indicates which of the two positions ('head' or 'manager') has a higher priority concerning room assignment. The screenshot as well as the results given below are generated using a prototypical implementation of the inference scheme. The prototype can be used to build models of uncertainty and calculate the valuations of single variables within the model.

To illustrate the inference process, we provide the results of the calculation performed by the system. The results are taken from test-outputs of the system while running the inference using the norms T_3 (maximum) and S_3 (minimum) for aggregation, T_2 (multiplication) for combination and again S_3 for marginalization We start with the calculation for the valuation $cert(HIGHER = head)$. The determination of this valuation is done by first aggregating the existing valuations of the different hypotheses (in this example we used dummy hypotheses h_i for the phenomena IMPORTANCE and HIERARCHY to minimize the space used for explanations). The result is immediately combined with the valuation of the causal relation. To avoid unnecessary calculations, this process is only triggered if a valuation for the causal influence of a configuration of hypotheses exists. The valuations used in our example are listed below together with the results they produce.

1. $cert(\wedge(h_1, h_1) \to head) = 0.9$

```
Configuration :11
Valuations  : 0.500000  0.700000
Aggregation : 0.500000
Causality   : 0.900000
Combination : 0.450000
```

2. $cert(\wedge(h_1, h_2) \to head) = 1$

```
Configuration :12
Valuations  : 0.500000  0.200000
Aggregation : 0.200000
Causality   : 1.000000
Combination : 0.200000
```

3. $cert(\wedge(h_1, h_3) \to head) = 0.5$

```
Configuration :13
Valuations  : 0.500000  0.100000
Aggregation : 0.100000
Causality   : 0.500000
Combination : 0.050000
```

Now the results of the three inference steps (0.45, 0.2 and 0.05) are marginalized using the maximum operator S_3. Therefore the valuation of the assertion *"The highest position in the research group is that of the head of the group."* represented as (HIGHER = head) in the model of uncertainty is 0.45 as calculated by the system.

```
Marginal : 0.450000
```

The corresponding valuation for the conflicting assertion *"The highest position in the research group is that of the manager"* represented as (HIGHER = manager) is calculated in the same way using the existing valuations of the causal relation.

1. $cert(\wedge(h_2, h_1) \rightarrow manager) = 0.9$

```
Configuration :21
Valuations : 0.100000  0.700000
Aggregation : 0.100000
Causality : 0.900000
Combination : 0.090000
```

2. $cert(\wedge(h_2, h_2) \rightarrow manager) = 0.5$

```
Configuration :22
Valuations : 0.100000  0.200000
Aggregation : 0.100000
Causality : 0.500000
Combination : 0.050000
```

3. $cert(\wedge(h_2, h_3) \rightarrow manager) = 1$

```
Configuration :23
Valuations : 0.100000  0.100000
Aggregation : 0.100000
Causality : 1.000000
Combination : 0.100000
```

The marginalization of the valuations determined above (0.09, 0.05, 0.1) results in the valuation cert(HIGHER = head) = 0.1 as calculated by the system. So the overall result produced by uncertain inference within the model of uncertainty are the valuations of the two competing hypotheses (HIGHER = head) and (HIGHER = manager).

```
Result:

HIGHER = head : 0.450000
HIGHER = manager : 0.100000
```

The question left open is how this result can be further used in the problem-solving process specified in the model of expertise. Espcially there is a need for a mechanism to transfer knowledge between the two models. Concerning our example it has to be specified how the intuitive result of the inference process that the head is the highest position can be formally integrated in the axiomatic model of expertise.

3.4 Transfer of Knowledge

To perform the determination of the assertional knowledge stated in section 3 the result of the uncertain inference has to be handed over to the model of expertise. As phenomena of uncertainty represent different hypotheses about the state of the world their meaning $sem(UP)$ consists of the meanings of these different possibilites given by the meaning of the hypotheses $sem(x_i)$ which are disjunctively aggregated to indicate that each one is possible. So the meaning of a phenomenon of uncertainty is given by an axiom that is defined as described in the following definition.

Definition: 18 (Axiom of uncertainty) *Let UP be a phenomenon of uncertainty with* $W_{UP} = \{x_1, \cdots, x_n\}$ *then its semantics* $sem(UP)$ *are given by an axiom of uncertainty of the form*

$$sem(UP) := \bigvee_{i=1}^{n} sem(x_i) \qquad (6)$$

where $sem(x_i)$ *denotes the sematics of the phenomenon, provided* x_i *is known to be the true hypothesis.*

Once we have defined the meaning of a phenomenon of uncertainty via the disjunction of the different hypotheses contained therein it is straightforward to do the same for the result of an inference process in the model of uncertainty. We just assume that the result is the selection of one hypothesis which is assumed to reflect the true state of the world. To simplify matters we define the hypothesis with the highest valuation to be the accepted one. More elaborated notions of acceptance can be found in [9]. We can now simply replace the meaning of the phenomenon of uncertainty by the meaning of the chosen hypothesis.

Fig. 3. Interface of the Models

Definition: 19 (Assumptions) *Let UP be a phenomenon of uncertainty and $x \in W_{UP}$, then an assumption may be made under the following condition*

$$assumption(UP, x) \Rightarrow V_{UP}(x) = max\{V_{UP}(y)|y \in W_{UP}\} \tag{7}$$

The impact of the assumption is given by replacing $sem(UP)$ by $sem(x)$ in the model of expertise.

The other way round, we can also use facts that can be derived from the model of expertise as evidence in the model of uncertainty. If the meaning $sem(x_i)$ of a single hypothesis x_i can be derived from the theory that contains the uncertainty axiom $sem(UP)$ of the corresponding phenomenon of uncertainty UP then the other hypotheses can be ruled out in the inference step. Technically this can be done by replacing the structure equation determining this phenomenon by the expression $UP = x_i$. This replacement corresponds to an atomar manipulation $do(UP = x)$ of the causal model as described by Pearl [13].

Definition: 20 (Evidences) *The equation $ref[theory](X)$ denotes the logical theory containing the axiom of uncertainty describing the phenomenon X. Let UP be a simple phenomenon of uncertainty and $x \in W_{UP}$, then an evidence is defined by*

$$evidence(UP, x) \equiv (ref[theory](UP) \vdash sem(x)) \tag{8}$$

The impact of the evidence is given by replacing the structural equation determining X by the equation: UP = x ($do(UP = x)$)

The concrete definition of the meaning of a single hypothesis again depends on the specification language that is used to describe the model of expertise.

4 Distributed Execution of Uncertain Inference

A more detailed analysis of the inference process using graph-based concepts reveals that there is some potential for optimizing the efficiency of the inference process using a distributed approach [20]. We omit the in-depth analysis of the system dynamics and concentrate on the description of a graph-based representation of the model of uncertainty that enables us to use virtual processors sited at the hyperedges of the resulting hypergraph performing a distributed update of hypothesis valuations. Doing this the processors strictly follow the inference scheme described in section 3.2.

4.1 Graph-based Interpretation of Uncertainty Models

Hypergraphs are a generalization of common graphs, where the number of nodes connected by an edge is not limited to to the number of two. So the so-called hyperedges may connect an arbitrary number of nodes, referred to as hypernodes. A directed version of hypergraphs and their use for knowledge representation is described in [8]. In this work, a hyperedge v_i is defined by two sets S_i and T_i

(which must not be mixed up with the notion of triangular norms !) where S_i is a set of nodes which are the sources of v_i and T_i a set of target nodes. The distintion between source and target determines the direction of the hyperedge.

Definition: 21 (directed hypergraph) *A directed hypergraph is a tuple $H = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ is a set of nodes and $E = \{E_1, \cdots, E_m\}$ a set of directed hyperedges, where $E_i = (S_i, T_i)$ with $S_i, T_i \subseteq V, S_i \cap T_i = \emptyset$ for $i = 1, \cdots, m$.*

Hypergraphs are able to represent arbitrary relations between objects by assigning a node to each object under consideration in such a way that relations between these objects are represented by hyperedges. In the case, where only functional relations are to be represented, the number of nodes in the target-set of each hyperedge may be limited to one. According to [8] the resulting hypergraph is called a backward-hypergraph or B-graph.

Definition: 22 (B-graph) *A backward-hypergraph (B-Graph) is a hypergraph $H = (V, E)$ with $|T_i| = 1$ for all hyperedges $E_i, i = 1, \cdots, m$.*

A backward-graph can be used to represent the model of uncertainty as defined in section 2.4. The phenomena of a model can be modelled as nodes in the hypergraph. As phenomena are connected through functional relations (structural equations) the relations used in the inference process may be modelled by hyperedges. Then the target-node of a hyperedge stands for the image of the function over the input-phenomena which are now elements of the set of source-nodes attached to each hyperedge. A graph-based interpretation of the model of uncertainty leads to the following alternative definition of a model of uncertainty.

Definition: 23 (Model of Uncertainty) *Let $\mathcal{UP}$ be the set of all phenomena, $\mathcal{OPS}$ the set of all operators and $\mathcal{UKS}$ the set of all unertain knowledge sources, then the model of uncertainy UNC formed by these concepts is a B-graph $H_{\mathcal{U}} = (V_{\mathcal{U}}, E_{\mathcal{U}})$, which is defined as*

(i) $V_{\mathcal{U}} = \mathcal{UP}$
(ii) $E_{\mathcal{U}} = \mathcal{OPS} \cup \mathcal{UKS}$

For each hyperedge v_i the set of source nodes is give by the set of all parameters of the corresponding causal function, while the result of this function serves as target node.

To differentiate between operations and uncertain knowledge sources it is admissible to type the hyperedges of the model according to the type of causal relation they refer to. Though the inference mechanism does not distinguish between conjunction operators and uncertain knowledge sources we decided to use different graphical representations for operators, uncertain knowledge sources,

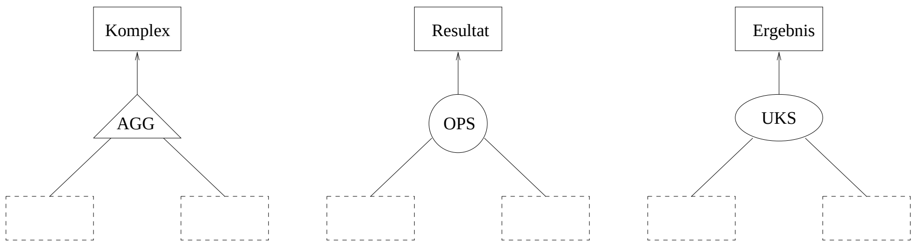

Fig. 4. Hyperedge-representation of the different causal relations

and so-called aggregations due to the conceptual difference (The concepts of aggregation is not considered in this paper. See [21, 14] for a detailed description). Figure 4 shows the different types of hyperedges used.

The generation of a graph-based model of uncertainty can be done in an iterative process. First of all the phenomenon that has to be determined is chosen to be the root node of the B-graph. After that the function determining this node and the parameters of this function are transformed into a hyperedge definition. This is repeated for each node in the source-set of the existing hyperedges that is also determined by a causal function, extending the model with further hyperedges until all phenomena and causal relations are modelled.

4.2 Representation of Valuation Functions

To realize a distributed execution of the inference process the involved valuations have to be stored in such a way, that they are always available at the place where they are needed. This causes no problems considering the valuations of hypotheses because they are a part of each phenomenon of uncertainty and are therefore sited at the nodes of the model. We chose to use vectors of real numbers to represent these valuation functions. Such a valuation-vector is attached to each phenomenon X in the model. The i-th place pi_i of the vector holds the valuation $cert(X = x_i)$. These vectors can be placed at the nodes representing the corresponding phenomenon of uncertainty.

Definition: 24 (valuation-vector) *The valuation function for the hypotheses of a phenomenon of uncertainty $X \in V_{\mathcal{U}}$ is given by a vector* $\mathbf{h} - \mathbf{valuation}$ *with*

$$\pi_i = cert(X = x_i)$$

Concerning valuations of causal relations the situation becomes more complicated due to the fact, that more than one phenomenon has to be taken into account. Therefore it is not naturally clear where these valuations are sited. To solve this problem we refer to the graph-based definition of the model of uncertainty. We can exploit the property of the model, that each phenomenon is connected to all other phenomena that directly influence its valuation by a hyperedge representing the corresponding causal relation. For this purpose, we assume conditional independence between the phenomena that directly results from the underlying structural equations (see [13]). As a consequence we may represent valuations of causal relations at the hyperedges of the model.

We can no longer use vectors to represent the valuation functions of causal relations because we have to consider valuations of different hypotheses and of different configurations of influencing parameters. To capture these two dimensions, we use a two-dimensional matrix of real numbers. This matrix is filled in such a way that the position m_ij (i-th column, j-th line) is filled with the valuation of the causal influence of the i-th configuration of the parameters on the hypothesis h_j. As already defined, we refer to this valuation as $cert(f_Y(x_{i_1}, \cdots, x_{i_n}) \rightarrow Y = y_j)$.

Definition: 25 (valuation-matrix) *The valuation function of a causal relation $f \in E_{\mathcal{U}}$ is represented by a matrix* $\mathbf{c-valuation}$ *with*

$$m_{ij} = cert(f_Y(x_{i_1}, \cdots, x_{i_n}) \rightarrow Y = y_j)$$

By guaranteeing that all valuations are available at the places where they are needed to perform uncertain inference it becomes possible to distribute the calculation process over the different parts of the model performing independent calculation at the hyperedges and exchanging the results via the nodes of the model.

4.3 Computation with Hyperedge-Processors

Due to the local character of the inference scheme, all calculation necessary to determine the hypotheses valuations of a phenomenon can be performed at the hyperedge whose target node is the phenomenon that has to be determined. For this purpose, we define virtual processors placed at each hyperedge in the model executing the three-step inference scheme described in section 3.2. All necessary valuation are provided by the valuation-vectors of the source nodes and the valuation-matrix stored at the edge. The edge-processors perform a local update of its target node which is defined as follows.

Definition: 26 (Update) *Let $Y \in V_{\mathcal{U}}$ be a phenomenon of uncertainty with hypotheses set W_Y, then the update of Y is defined as*

$$\pi_i := cert(Y = y_i), y_i \in W_Y$$

where $cert(Y = y_i)$ is calculated using the inference scheme described above.

While this update process is equivalent for all phenomena, different results are caused by the different contexts of the different phenomena. In our model this context is given by the valuation of source nodes and hyperedge where the calculation is performed. This equivalence is the reason, why we can use standardised processors for the update of the model. Figure 5 shows the scheme of

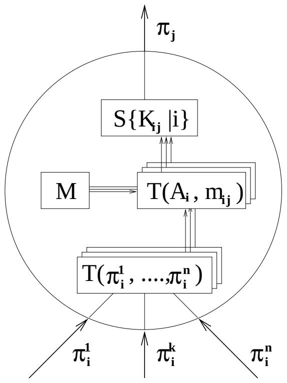

Fig. 5. Edge-processor used for distributed calculation of valuations

an edge-processor used in our model.

While the edge-processors only perform local computation, the overall inference mechanism is guided by a message passing mechanism between the hyperedges. If a processor is asked to give the valuation of its target-node it looks for the valuations that are necessary to perform the computation and triggers other edge-processors or asks for user input if necessary. This mechanism does not only results in a higher efficiency of the inference process but also provides a kind of user guidance[1] by querying the user or expert on demand.

5 Application of the Model

The approach is to be seen as a first step towards a methodology for handling uncertain knowledge in the context of an engineering process leading from acquisition and informal modelling of uncertain knowledge to a formal model of uncertain expertise which will serve as a basis for an operationalisation. In the course of an engineering methodology there are at least two direct applications of our approach that will be briefly presented in the following paragraphs.

5.1 Formalisation of Uncertain Reasoning

The model described above has been used to develop a specification language for the representation of expertise under uncertainty [19]. This language called FLUE (Formal Language for Uncertain Expertise) is an extension of the formal specification language $(ML)^2$ [22], which has been mentioned briefly in section 2.

To integrate uncertainty aspects smoothly into the existing parts of the language, a textual description of our model of uncertainty has been developed

[1] One can even think of an automatical generation of user interfaces

which is based on the semantical mapping defined from this model into the theories of the logics underlying $(ML)^2$. These theories consist of a signature describing the language used and a set of axioms. The overall structure of a specification is built up via import-relations.

The language FLUE adopts this scheme. Each phenomenon of uncertainty is specified in a theory. The signature of this theory is given by concepts from the model of expertise the phenomenon refers to. Instead of logical sentences, the set of axioms defines a valuation function over the hypothesis space of the phenomenon and the structure equation determining the phenomenon. The connection to the relevant context, which is a parameter of the structure equation is established through import-relations as used in $(ML)^2$.

5.2 Engineering of Causal Networks

The causal nature of the model of uncertainty comes along with some very beneficial possibilities that arise from the properties of causality that enable an efficient operationalisation. There is a wide range of calculi which can be used for this purpose, because inference in the model is equivalent to inference in a valuation-based system [16]. Therefore existing approaches for describing well-known calculi of uncertainty in terms of a valuation-based system can be used to derive the valuations required. Examples for such calculi are [18], [4].

In connection with subjective probabilities causal networks are widely used for the implementation of uncertain inference [12]. There are also professional tools available that support creation and use of these so-called Bayesian networks [1]. Thus, provided all valuations used are probabilities, our model of uncertainty can easily be transformed into an executable Bayesian network. From this point of view our model of uncertainty serves as an interface between a model of expertise and a Bayesian network. The model adopts the structure specified in the model of expertise via the reference mapping. It therefore provides a scheme which can be instantiated by a Bayesian network. This process facilitates the layout of the network because the knowledge engineer is able to concentrate on the description of the problem-solving process instead of a direct encoding of expertise which will necessarily raise trouble due to the limited expressiveness of Bayesian networks.

6 Conclusion

We have presented an inference scheme, that can be used to reason within a model of uncertainty describing uncertain parts of a model of expertise. The inference scheme can be used to bridge gaps in logic-based specifications that arise from uncertainty concerning assertional knowledge by using sets of hypotheses with attached valuations.

Our instantiation of the inference scheme can be seen as a straightforward possibility to perform reasoning on a higher level of abstraction than usually needed in knowledge-based systems. Therefore the aim of this scheme is merely to provide prototyping facilities that support the knowledge engineering process. While the static model of uncertainty decribed in section 2 provides the basis for specification languages that contain explicit notions of uncertainty [19] supporting the acquisition and specification of uncertain knowledge, the dynamic part of the model which has been described in section 3 provides the possibility to create operational specifications. Based on the notion of valuation-based systems the model of uncertainty provides an intermediate step towards the use of concrete calculi for reasoning under uncertainty.

A connection to concrete calculi like the ones available in the PULCINELLA-Tool [15] can be established by translating the model of expertise into the representation of the chosen calculus. In this setting, reasoning is performed in the concrete calculus using the structure of the model of expertise. The connection between the calculus and the model of expertise is done by reinterpreting the used calculus in terms of a VBS (such reinterpretations have already been developed for probability [18], possibility [4] and evidence theory [16]). The model of uncertainty now serves as an interface between logic-based models of expertise and the graph-based approaches for reasoning under uncertainty.

There are several directions for further research. We only want to suggest two of these directions that are directly concerned with the combination of specification languages and a causal model of uncertainty.

Executable Specifications play a very important role in knowledge engineering because they can be used to build a prototypical system without loosing the advantages of (formal) model-based approaches. While our model of uncertainty can easily be operationalized as discussed in the previous section, up to now our model does not support executable specifications, because the model of expertise we used to develop the language FLUE is not completely executable. Therefore one topic for further research is to connect the model of uncertainty to an executable specification language to achieve an executable specification containing the whole model of uncertain expertise.

Problem-Solving Methods have become of major interest in the knowledge engineering community for several reasons. Maybe the most important are that they (1) seem to bridge the gap between specification and design of knowledge-based systems and (2) form the most promising components to be reused across different applications reducing the development costs for knowledge-based systems. Therefore the second direction for further research is to apply the ideas of the approach presented in this paper to languages that focus on the description of problem-solving methods. In contrast to the use of executable specification languages, this will require much more investigation because the terminology described in section 2.1 is no longer adequate. Furthermore it is not clear if a static model of uncertainty will work in the context of problem-solving methods. The character of problem-solving methods raises questions towards dynamic modifications of the model of uncertainty.

References

1. S.K. Andersen, K.G. Olsen, F.V. Jensen, and F. Jensen. HUGIN: A Shell for Building Bayesian Belief Networks for Expert Systems. In *Proceedings of the 11th International Joint Conference on Artificial Intelliegence (IJCAI-89)*, pages 1080 – 1085, San Mateo, CA, 1989. Morgan Kaufmann Publisher.
2. Juergen Angele, Stefan Decker, Rainer Perkuhn, and Rudi Studer. Modelling Problem-Solving Method in NewKARL. In *Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW96)*, Banff, Canada, 1996.
3. P. Bonissone and K. Decker. Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-Off Precision and Complexity. In J.F. Lemmer L.N. Kanal, editor, *Uncertainty in Artificial Intelligence*. North-Holland, 1986.
4. D. Dubois and H. Prade. Inference in Possibilistic Hypergraphs. In *Procs. of the 6th IPMU Conference*, Paris, 1990.
5. D. Fensel and F. van Harmelen. A comparison of languages which operationalise and formalise KADS models of expertise. *The Knowledge Engineering Review*, 9:105–146, 1994.
6. Dieter Fensel, V. Richard Benjamins, Stefan Decker, Mauro Gaspari, Rix Groenboom, W. Grosso, Enrico Motta, Enric Plaza, Guus Schreiber, Rudi Studer, and Bob Wielinga. A Unified Problem-Solving Method Description Language. A draft version can be achieved from the first authors home page.
7. Dieter Fensel, Rix Groenboom, and G. R. Renardel de Lavalette. Modal Change Logic (MCL): Specifying the Reasoning of Knowledge-based Systems. *Data and Knowledge Engineering*, 26(3):243–269, 1998.
8. G. Gallo and G. Longo. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42:257 – 278, 1993.
9. H.E. Kyburg. Probabilistic Acceptance. In *Uncertainty in Artificial Intelligence*, 1997. Proceedings of the 13th International Conference.
10. Enrico Motta. KBS modelling in OCML. In *Proceedings of the 5th Workshop Knowledge Engineering Methods and Languages (KEML'95)*, 1995.
11. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Series in Representaion and Reasoning. Morgan Kaufmann, San Mateo, Ca, 1988.
12. J. Pearl. From Bayesian Networks to Causal Networks. In A. Gammerman, editor, *Bayesian Networks and Probabilistic Reasoning*. Alfred Walter Ltd., London, 1994.
13. J. Pearl. Structural and Probabilistic Causality. In D.R. Shanks, K.J. Holyoak, and D.L. Medin, editors, *The Psychology of Learning and Motivation*, volume 34: Causal Learning, pages 393–435. Academic Press, San Diego, CA, 1996.
14. K. C. Ranze and H. Stuckenschmidt. Modelling Uncertainty in Expertise. Proceedings of IT&KNOWS Conference, 15th IFIP World Computer Congress '98.
15. A. Saffiotti and E. Umkehrer. PULCINELLA - A General Tool for Propagating Uncertainty in Valuation Networks. In *Proc. of the 7th Conf. on Uncertainty in AI*, pages 323 – 331, Los Angeles, CA, 1991.
16. P.P. Shenoy. Valuation-based Systems: A Framework for Managing Uncertainty in Expert Systems. In *Fuzzy Logic for the Management of Uncertainty*. Wiley and Sons, 1989.
17. P.P. Shenoy. Consistency in Valuation-Based Systems. *ORSA Journal on Computing*, 6(3):281 – 291, Summer 1994.

18. P.P. Shenoy and G. Shafer. An Axiomatic Framework for Bayesian and Belief-Function Propagation. In *Procs. of AAAI Workshop on Uncertainty in AI*, pages 307 – 314, 1988.
19. H. Stuckenschmidt and K.C. Ranze. FLUE : A Specification Language for Uncertain Expertise. Technical report, Center for Computing Technology, University of Bremen, 1998. to appear.
20. Heiner Stuckenschmidt. Specification of uncertain knowledge in a formal model of expertise. Master's thesis, TZI, University of Bremen, Germany, 1998. (in german).
21. Heiner Stuckenschmidt and K. Christoph Ranze. Capturing uncertainty in models of expertise. In *Proceedings of the KEML98 workshop*, Karlsruhe, Germany, 1998.
22. F. van Harmelen and J. R. Balder. $(\mathrm{ML})^2$: A formal language for KADS models of expertise. *Knowledge Acquisition Journal*, 4(1), 1992. Special issue: 'The KADS approach to knowledge engineering', reprinted in KADS: A Principled Approach to Knowledge-Based System Development, 1993, Schreiber, A.Th. et al. (eds.).
23. F. van Harmelen, R. Lopez de Mantaras, J. Malec, and J. Treur. Comparing formal specification languages. In J. Treur and Th. Wetter, editors, *Formal Specification of Complex Reasoning Systems*, Workshop Series, pages 257–282. Ellis Horwood, 1992.
24. Frank van Harmelen. Definable naming relations in meta-level systems. In A. Pettorossi, editor, *Proceedings of the Third Workshop on Meta-programming in Logic (META'92)*, volume 649 of *Lecture Notes in Computer Science*, pages 89–104, Uppsala, June 1992. Springer-Verlag.
25. B. Wielinga, W. Van de Velde, G. Schreiber, and H. Akkermans. Expertise Model Definition Document. Technical Report KADS-II/M2/UvA/026/1.1, University of Amsterdam, Free University of Brussels, Netherlands Energy Research Foundation ECN, May 1993.
26. B. Wielinga, A. Schreiber, and J. Breuker. KADS: A Modelling Approach to Knowledge Engineering. *Knowledge Acquisition*, 4(1):5–53, 1992.