Validation and verification of conceptual models of diagnosis

van Harmelen, F.A.H.; ten Teije, A.

published in
Validation and verification of conceptual models of diagnosis
1997

Proceedings of the Fourth European Symposium on the Validation and Verification of Knowledge Based Systems (EUROVAV'97), pp. 117--128, 1997, Leuven, Belgium

Validation and verification of conceptual models of diagnosis

Frank van Harmelen

Dept. of Math and CS
Vrije Universiteit Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
frankh@cs.vu.nl

Annette ten Teije[1]

SWI
University of Amsterdam
Roetersstraat 15
1018 WB Amsterdam
The Netherlands
annette@cs.vu.nl

Abstract: Traditional approaches to validation and verification of KBS aim at investigating properties of a KBS which are *independent of the particular task* of the KBS, and are phrased in terms of the *implementation language* of the final system. In contrast to this, we propose an approach to validation and verification of KBS which exploits *task-specific properties* of a KBS, and which is based on an *implementation-independent conceptual model* of the system. We illustrate our approach by investigating the task-specific properties of a conceptual model for a wide class of diagnostic systems.

Keywords: Conceptual Model, Diagnosis, Formal Modelling, Task-specific properties

[1]Currently, she is carrying out her work at Imperial Cancer Research Fund (London) as part of a community training project (TMR) financed by the European Commission. Project: ERBFMBICT961130

1 Introduction

1.1 Motivation

Much of the current work on V&V of knowledge-based systems has the following characteristics: Firstly, it is aimed at verifying properties of the *implementation* of a KBS (e.g. in rules or frames). Secondly it verifies properties which are *independent of the particular task* of the KBS (e.g. loop-freeness of a rule-set, absence of redandant rules, etc). Thirdly, it verifies properties of the *knowledge* of a KBS, but ignores the *control strategy* that is employed (e.g. forward or backward reasoning, exhaustive or non-exhaustive, etc).

In this paper, we propose a complementary approach to V&V of KBS, namely: (i) we verify properties that are *independent of a particular implementation formalism*, but that are based on a more abstract analisys of the knowledge involved in a KBS; (ii) we verify properties which *exploit the characteristics of the specific task* of the KBS; (iii) the verification of these properties is *based on the particular reasoning strategy* that is employed to use this knowledge.

This complementary approach to existing V&V techniques is required for the following reasons:

Multiple models: modern Knowledge Engineering techniques proceed through a series of models [Wielinga *et al.*, 1992] of which implementation is one of the last in the process. One would like to do V&V on earlier models. In particular the Conceptual Model plays an important role in the development process.
All this is similar to the Software Engineering life cycle (such as requirement analysis, specification, design, implementation), where we would like to do V&V in earlier stages than the implementation. (see [Fensel, 1995] for a comparision of the modelling techniques from Software Engineering and Knowledge Engineering).

Repair strategies: Current V&V techniques give no prescription on how to repair a fault once found. For example, there are multiple ways of repairing a dead-end rule (e.g. removing the rule, or adding a condition to some other rule). Current techniques do not tell us how to choose between these different repairs. This absence of repair-prescriptions is due to the very generic nature of the properties identified by current V&V techniques (which are completely independent of the particular task of the specific KBS that is being validated). Once we verify properties that are task specific, we can verify against a specification of the intended behaviour of the KBS, and consequently also prescribe repair strategies.

In this paper, we will illustrate our general approach using a conceptual model for a wide class of diagnostic systems. For such diagnostic systems, the following are examples of the task specific properties that we will investigate: definition of what a diagnosis is, maximal size of a diagnosis (e.g. single faults), maximal number of alternative diagnosis (e.g. unique diagnosis), preference on diagnoses (e.g. only the most urgent diagnoses), and thresholds on diagnoses (e.g compute only diagnoses with a minimal reliability).

Although we will illustrate our approach on diagnostic systems, our approach is by no means limited to such systems. Similar properties can be identified for other types of KBS, for example for planning systems.

1.2 Related work

In the Knowledge Engineering community, much work has been done on task specific conceptual models, but very little has been done on deriving properties of such models. Examples of work dealing with deriving properties of conceptual models are the following:

[vanHarmelen & Aben, 1995] already proposes to do validation and verification on the basis of conceptual models, but still only studies task-independent properties of conceptual models. As a consequence this work suffers from some of the same problems as the traditional V&V approach (e.g. lack of repair strategies).

[Benjamins, 1994] does indeed formulate task dependend properties of conceptual models, but does so using only informal descriptions of both the conceptual model and of the properties.

Consequently, this work does not lend itself as the basis for a sufficiently formal approach to validation and verification.

The work in [Fensel *et al.*, 1996] is very close to our aims (namely the formal derivation of task dependent properties of a conceptual model), but has so far only been applied to small examples. Furthemore, it puts more emphasis on the dynamic and algorithmic behaviour of a KBS, whereas we focus on the declarative I/O specificiation of a KBS.

Recent work by [Cornelissen *et al.*, 1997] is again very similar in goals to ours, but uses a different method for proving properties. They use the hierarchical decomposition structure of a conceptual model as a basis for their proofs. We will use a non-hierarchical decomposition for the same purpose. They have also applied their techniques to diagnostic conceptual models, but our conceptual models and our properties are closer to realistic diagnostic systems.

Summarising, our goals are as follows:

- Verification and validation of conceptual models instead of implementations of KBS.
- Verification of task-specific properties.
- Give prescriptions on how to change the conceptual model when a desired property fails to hold.

In the remainder of this paper, we shall first present our conceptual model of diagnostic systems (section 2). We shall then show how this model can be used for verification (section 3) and validation (section 4) of task-specific properties. In section 5 we will illustrate our approach in a small example.

2 A conceptual model for diagnosis

In general, a diagnostic problem arises if there is a discrepancy between the observed behaviour of a system (e.g. an artifact) and how the system should behave, in other words, the expected behaviour does not correspond with reality. The diagnostic task is to find out the cause of this discrepancy. A diagnostic method computes the solutions for a diagnostic problem by using a model of the expected behaviour (the behaviour model, *BM*), the actually observed behaviour *OBS*, and contextual information *CXT*. The computed solutions of a diagnostic problem represent an explanation for the observed behaviour.

Our conceptual model of diagnostic problem solvers is based on the following general account of their functionality: An explanation distinguishes *two types of observations*: it covers some observations, and it does *not contradict* other observations. The explanation is restricted to a *vocabulary* of special candidates that could be causes of a behaviour discrepancy (e.g. components). Usually we are not interested in all possible explanations, but only the *most reasonable* explanations. We also want to *represent* an explanation as a solution that a user can interpret. (For example, in medical domains, users are usually interested in the disease, and not in all the current states of the parts of the patient's body).

Together, these six aspects written in italics make up the particular notion of diagnosis that is realised in a given method. We can capure these general characteristics of a diagnostic method in the following formal definition:

When given as input the behaviour model *BM*, a context *CXT* and a set of observations *OBS*, a diagnostic method computes a set of solutions *Sol* such that:

Definition 1 (Diagnosis)

$$\begin{array}{llll}
\underline{\text{OMap}}(OBS) = \langle Obs_{cov}, Obs_{con} \rangle & & & and \\
\overline{Es} = \{E | & BM \cup E \cup CXT \vdash_{\underline{cov}} Obs_{cov} & and & \\
 & BM \cup E \cup CXT \not\vdash_{\underline{cov}} \bot & and & \\
 & BM \cup E \cup CXT \not\vdash_{\underline{con}} \neg Obs_{con} & and & \\
 & E \subseteq \underline{Voc}\} & & and \\
\underline{\text{Select}}(Es, E') & & & and \\
\underline{\text{Solform}}(E', Sol) & & &
\end{array}$$

Each of the six underlined terms is a parameter in our representation of diagnostic methods. Varying one or more parameters amounts to describing a different diagnostic method. The observation-mapping *OMap* determines which observations must be explained (or: covered) Obs_{cov}, and which need only not be contradicted (Obs_{con}). *E* is an explanation for the observed behaviour by covering some observations ($\vdash_{cov}$), and not contradicting others ($\not\vdash_{con}$). We write $\vdash_{cov}$ and $\not\vdash_{con}$ as different symbols to emphasise that one is not necessarily the negation of the other, and that neither is necessarily the same as the classical entailment $\vdash$. *E* is expressed in a particular vocabulary Voc. We are interested in the most reasonable explanations, determined by a selection criterion *Select*. The form of the solution (*Solform*) determines the representation of the final result of the method. We will use the term *explanations* for the set Es. Since the notion of an explanation is determined by the first four parameters, we will write

$$Es = explanations(\mathit{OMap}, \vdash_{cov}, \vdash_{con}, Voc)$$

as a shorthand for the set of explanations computed by the first part of the diagnostic model [1], and

$$diagnoses(\mathit{OMap}, \vdash_{cov}, \vdash_{con}, Voc, \mathit{Select}, \mathit{Solform})$$

as a shorthand for the entire conceptual model of Def. (1).

The dependencies between all these components of a diagnostic method is shown in figure 1. This graphical representation of our conceptual model illustrates that the model does in fact correspond to the inference layer of a full KADS model [Wielinga *et al.*, 1992]. Such models are widely used in the Knowledge Engineering community, both in research and in applications.

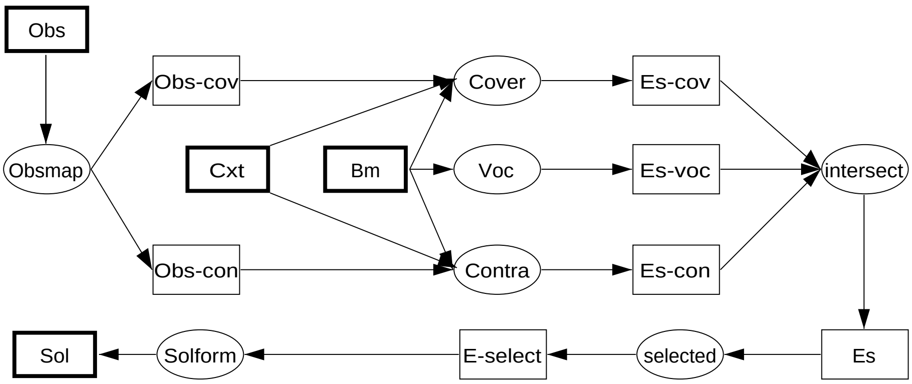


Figure 1: Components of diagnostic methods and their relations. Ovals are components, boxes are their inputs/outputs, thick boxes are inputs/outputs of the entire method

In [tenTeije, 1997] we have shown that this model is sufficiently general to capture a wide class of diagnostic models, including such varying types of diagnosis as: a pure abductive system, an abductive system with preferences or abstractions, a set-covering system, a pure consistency based system, a consistency based system with abstractions, or with fault modes, or with preferences.

It is important to notice that this definition of a conceptual model for diagnosis introduces a *task-specific ontology*. This task-specific ontology will be the basis for the verification and validation of properties in the subsequent sections. This is in sharp contrast with results from traditional work on V&V of KBS, which is typically framed in a task-independent ontology (for instance in terms of premises and conclusions of rules). This introduction of a task-specific ontology is a crucial step in our attempt to tackle more task-specific properties in the verification process.

[1] In this paper we freely mix first-order and higher-order notation, quantifying over higher order paramaters such as *OMap*. In [tenTeije & vanHarmelen, 1996c] we have carefully formalised all the notions from this paper in a three-layered meta-architecture, using only first-order logic in each of the layers.

3 Verification of properties

In this section, we deal with properties that are required for a legal or meaningful conceptual model. Any conceptual model that fails any of these properties does not qualify as a meaningful model of a diagnostic method.

3.1 Type information

Input and output of the all components must be of the right type, for example *Select* must be a relation between a set of explanations and a single explanation:

$$\textit{Select} : \textit{set of explanations} \times \textit{explanation},$$

and *OMap* must be a mapping from a set of observables to a binary tuple of sets of observables:

$$\textit{OMap} : \textit{set of observables} \mapsto \langle \textit{set of observables, set of observables} \rangle$$

We will not further elaborate on this rather obvious check on the correctness of specification of a conceptual model, except to say that these and other type-correctness properties can be used to verify that the conceptual model is a meaningful diagnostic model.

3.2 Observation Mapping *OMap*

The *OMap* parameter determines which observations must be covered by the explanantion, and which need only be consistent with the explanantion. We refer to definition 3 for a number of examples of this mapping. Two requirements on this parameter are that firstly it assigns a role to at least all the given observations in OBS (ie no observations may be ignored):

$$\begin{array}{ll} \text{If} & \textit{OMap}(OBS) = \langle O_{cov}, O_{con} \rangle \\ \text{then} & OBS \subseteq (O_{cov} \cup O_{con}) \end{array}$$

and secondly that the division of observations is uniquely determined by the given observations OBS, ie *OMap* must be a function:

$$\textit{OMap}(OBS_1) \neq \textit{OMap}(OBS_2) \rightarrow OBS_1 \neq OBS_2$$

Again, as with all properties in this section, these properties must hold for the conceptual model to be a meaningful diagnostic model.

3.3 The consequence relations $\vdash_{cov}$ and $\vdash_{con}$

Typically, the standard notions of first order entailment and consistency are used to model the notions of diagnostic of covering and consistency. However, a large number of other entailment relations is available in the literature. There is no clear notion of what properties an entailment entailment $\vdash$ should minimally satisfy, but [Hacking, 1994] suggests reflexivity, ($A \vdash A$), transitivity (if $A \vdash B$ and $B \vdash C$ then $A \vdash C$) are suggested as minimal. Other properties which might be considered (but which do not hold for many non-classical deduction relations) are monotonicity (if $A \vdash B$ then $A \wedge C \vdash B$) and dilution (if $A \vdash B$ then $A \vdash B \vee C$), or weaker versions of monotonicity such as: if $A \vdash B$ and $A \vdash C$ then $A \wedge C \vdash B$.

Although no widely accepted minimal conditions on consequence relations exist, any particular set of restrictions that turns out to be useful for a particular application can be enforced as part of the verification of the conceptual model.

Whatever values of $\vdash_{cov}$ and $\vdash_{con}$ we choose, it will always be necessary to obey the conceptual requirement that covering represents a stronger notion of explanation than consistency:

$$(\phi \vdash_{cov} \psi) \rightarrow (\phi \not\vdash_{con} \neg\psi)$$

This property follows from the stronger but natural requirement:

$$(\phi \vdash_{cov} \psi) \rightarrow (\phi \vdash_{con} \psi)$$

3.4 The explanatory vocababulary Voc

Only few requirements can be imposed on the choice of vocabulary of explanations, except that the vocabulary must be uniquely determined by the behaviour model, that the vocabulary must be contained in this behaviour model, and that observables are not allowed to explain themselves:

$$\begin{array}{l} Voc(BM_1) \neq Voc(BM_2) \rightarrow BM_1 \neq BM_2 \\ Voc(BM) \subseteq literals(BM) \\ Voc(BM) \cap observables = \emptyset \end{array}$$

Notice that the use of the set of possible $observables$ is another example of the use of the task-specific ontology that we introduce to formulate the particular properties that we verify.

3.5 The selection component *Select*

For a meaningful component that selects prefered explanations among all possible explanations, we demand that it indeed only does selection, ie:

$$\mathit{Select}(Es, E) \rightarrow E \in Es, \qquad (1)$$

and that it does do some form of selection (ie. it is not trivial):

$$\exists E, Es : E \in Es \wedge \neg \mathit{Select}(Es, E).$$

This second requirement is equivalent to disallowing the other direction of (1).

Furthermore, we demand that *Select* is defined for all possible sets of explanations:

$$Es = explanations(\mathit{OMap}, \vdash_{cov}, \vdash_{con}, Voc) \rightarrow \exists E : \mathit{Select}(Es, E)$$

Examples of often used values for the selection component are subset-minimality (selecting only those explanations for which no subset is also an explanation) and cardinality-minimality (selecting only those explanations whith the smallest number of elements):

Definition 2 (Selection criteria)

$$\begin{array}{rcl} \subseteq\text{-}min(Es, E) & \leftrightarrow & \neg \exists E' \in Es : E' \subset E \\ \#\text{-}min(Es, E) & \leftrightarrow & \neg \exists E' \in Es : |E'| < |E| \end{array}$$

Notice that both of these selection criteria satisfy all the requirements listed above.

3.6 The representation of solutions *Solform*

As with the selection component, we demand that a solution form exists for any explanation that is generated by the preceding components:

$$\begin{array}{l} Es = explanations(\mathit{OMap}, \vdash_{cov}, \vdash_{con}, Voc) \wedge EsSel = \{E | \mathit{Select}(Es, E)\} \\ \rightarrow \\ \exists Sol : \mathit{Solform}(EsSel, Sol) \end{array}$$

4 Validation of properties

In this section we discuss properties of a conceptual model that can be used to capture the intended functionality of a diagnostic KBS. We will show how such properties can be used to validate the functionality of a KBS, and how our knowledge of these properties can be used to adjust the conceptual model if the KBS fails to satisfy the intended properties.

4.1 Properties following from *OMap*

The following gives some examples of different reasonable definitions of the *OMap* component, which maps observations into those that must be explained and those that must not be contradicted:

Definition 3 (Definitions of *OMap*)

omap_1	*encodes the assumption that any observable not known to be true can be taken as false* $O_{cov} = OBS$ $O_{con} = \{\neg o \mid o \in observations \setminus OBS\}$
omap_2	*corresponds to abductive diagnosis (all observations must be covered)* $O_{cov} = OBS$ $O_{con} = \emptyset$
omap_3	*corresponds to consistency-based diagnosis in the sense of [Reiter, 1987]* $O_{cov} = \emptyset$ $O_{con} = OBS$
omap_4	*requires that all abnormal findings must be covered and none of the normal findings must be contradicted* $O_{cov} = \{o \in OBS \mid abnormal(o)\}$ $O_{con} = \{o \in OBS \mid normal(o)\}$
omap_5	*requires that all observed symptoms must be covered, and no unobserved symptom must be contradicted* $O_{cov} = \{o \in OBS \mid positive(o)\}$ $O_{con} = \{o \in OBS \mid negative(o)\}$

We will now formulate a theorem that we can use during validation if it turns out that a diagnostic KBS produces too many explanations. For the above definitions, if we define the following ordering:

Definition 4 (Ordering on *OMap*'s)

$$\begin{array}{ccccccc} \text{omap}_1 & \prec & \text{omap}_2 & \prec & \text{omap}_4 & \prec & \text{omap}_3 \\ & & \text{omap}_2 & \prec & \text{omap}_5 & \prec & \text{omap}_3 \end{array}$$

then the following theorem holds:

Theorem 5 (A stronger *OMap* gives more solutions)

$$\text{OMap} \prec \text{OMap}' \rightarrow (explanations(., \text{OMap}, .) \subseteq explanations(., \text{OMap}', .))^2$$

This theorem tells us that if we want to reduce the number of explanations, then we must replace the current definition for the observation-mapping by a definition which is lower in the ordering.

However, under some conditions, the inclusion is actually an equality (so changing the *OMap* has no influence on the number of solutions):

Theorem 6 (A stronger *OMap* under negation-as-failure has no effect))

$$\begin{array}{ll} \text{OMap} = \langle O_{cov}, O_{con} \rangle & \wedge \\ \text{OMap}' = \langle O'_{cov}, O'_{con} \rangle & \wedge \\ (O_{cov} \cup O_{con}) = (O'_{cov} \cup O'_{con}) & \rightarrow \end{array}$$
$$explanations(..., \text{OMap}, \vdash_{NAF}, \vdash_{NAF}, ...) = explanations(..., \text{OMap}', \vdash_{NAF}, \vdash_{NAF}, ...)$$

[2] we will write $explanations(..., X, ...)$ and $explanations(..., X', ...)$ as a shorthand notation for two notions of explanation which only differ in component X.

This is so because for negation as failure we have

$$(\phi \vdash \neg\psi) \leftrightarrow (\phi \not\vdash \psi)$$

or equivalently

$$(\phi \vdash \psi) \leftrightarrow (\phi \not\vdash \neg\psi).$$

The left-hand side of this corresponds exactly to the use of $\vdash_{NAF}$ as $\vdash_{cov}$, and the right-hand side corresponds to the use of $\vdash_{NAF}$ as $\vdash_{con}$.

Thus, whereas theorem 5 told us that moving to an *OMap* definition which is lower in the ordering might reduce the number of alternative explanations, theorem 6 puts bounds on this result by stating that given the value of some other parameters (in this case $\vdash_{cov}$ $\vdash_{con}$) the number of alternative explanations actually remains constant when moving down in the ordering of observation mappings.

4.2 Properties following from $\vdash_{cov}$ and $\vdash_{con}$

In [tenTeije & vanHarmelen, 1996a] we have used the work on approximate deduction relatations from [Schaerf & Cadoli, 1995] to define approximate notions of diagnosis, by using their approximate deduction relations as values for the parameters $\vdash_{cov}$ and $\vdash_{con}$. We will not report on the details of this work. For the purposes of this work it suffices to know that an ordering can be defined on entailment-relations, on the basis of the soundness or completeness of the deduction relation:

Definition 7 (Ordering on $\vdash$)

$$(\vdash \prec \vdash') \leftrightarrow (\phi \vdash \psi \rightarrow \phi \vdash' \psi)$$

Thus, some approximate entailment relations $\vdash'$ are sound but incomplete approximations of the classication entailment relation $\vdash$ (namely when $\vdash \prec \vdash'$), while other approximations are complete but unsound (namely when $\vdash' \prec \vdash$). In [tenTeije & vanHarmelen, 1996a] we have shown how such approximate entailment relations can be used to influence the size, number and quality of various notions of diagnosis. Some of these results are summarised below:

Theorem 8 (Weakening an unsound $\vdash_{cov}$ gives larger abductive solutions))

$$\begin{array}{ll} \vdash_{cov} \succ \vdash'_{cov} \wedge & E \in explanations(..., \text{omap}_2, \vdash_{cov}, ...) \\ \rightarrow & \\ \exists E' \supseteq E: & E' \in explanations(..., \text{omap}_2, \vdash'_{cov}) \end{array}$$

If we take for $\vdash_{cov}$ the classical entailment relation, than this theorem says that any abductive diagnosis obtained using an unsound deduction relation is contained in a classical diagnosis.

We can exploit this theorem to reduce the size of individual diagnoses. Notice that this is different from theorem 5 which affected the number of alternative diagnoses. In [tenTeije & vanHarmelen, 1996b] we have shown examples of the use this: if a systems yields only very large diagnoses, this might prevent us from taking specific action on the diagnosis, since not all causes can be treated simultaneously. Strengthening the $\vdash_{cov}$ component to be more unsound will yield smaller diagnoses. In [tenTeije & vanHarmelen, 1996b] we have shown how such reductions can be obtained by retaining only the most urgent or the most dangerous of the contributing causes.

A similar theorem on using incomplete definitions of $\vdash_{cov}$ enable us to increase the number of alternative diagnoses. This can be useful if no diagnosis can be found at all. By including less reliable parts of the behaviour model, it becomes possible to increase the chance of finding a diagnosis after all.

4.3 Properties following from Voc

The choice of vocabulary influences the number of competing diagnoses which are computed: a smaller vocabulary may lead to a reduction in diagnoses:

Theorem 9 (Smaller vocabulary gives fewer explanations)

$$Voc \subseteq Voc' \rightarrow explanations(..., Voc, ...) \subseteq explanations(..., Voc', ...)$$

4.4 Properties following from *Select*

As for the components *OMap*, $\vdash_{cov}$ and $\vdash_{con}$ it is possible to define an ordering on values for this parameter. If we define the obvious ordering on selection criteria:

Definition 10 (Ordering on selection critera)

$$\text{Select} \prec \text{Select}' \leftrightarrow (\text{Select}(Es, E) \rightarrow \text{Select}'(Es, E))$$

then the following theorem follows immediately:

Theorem 11 (Stronger selection criteria give fewer explanations)

$$\text{Select} \prec \text{Select}' \rightarrow diagnoses(..., \text{Select}, ...) \subseteq diagnoses(..., \text{Select}', ...)$$

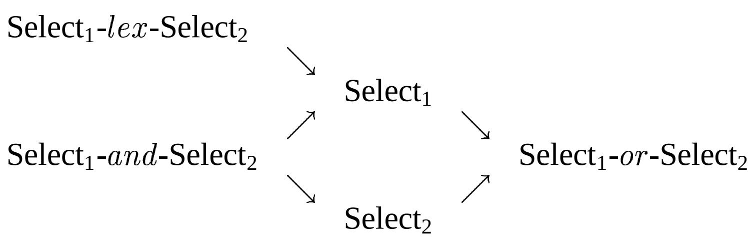
As before, we can exploit this theorem to reduce the number of diagnoses if during validation it turns out that the KBS produces too many competing diagnoses.

An example of such an ordering between specific selection criteria is

$$\#\text{-}min \quad \prec \quad \subseteq\text{-}min$$

Individual selection criteria can be combined to form composite selection criteria, for instance by taking the lexicographic, conjunctive or disjunctive combinations of the individial criteria. The orderings betweeen individial selection criteria induce an ordering between these composite selection criteria:

Theorem 12 (Orderings between composed selection criteria)
If we write Select $\rightarrow$ Select$'$ *as a graphical representation of* Select $\prec$ Select$'$*, then:*



Again, these results can be used to adjust the selection criterion of the diagnostic model if during validation it turns out that too many competing diagnoses are computed by the system.

5 An example

In many diagnostic applications, it is important to aim for the smallest number of competing diagnoses (preferably only one, unique diagnosis). This is important because a large number of competing makes it impossible to decide wich therapy or repair to select, since not all competing diagnoses can be repaired or treated simultaneously (for reasons of cost or because of interactions among the alternative treatments).

The theorems from the preceding section (section 4) allow us to validate this desired property in terms of the parameters that are used to instantiate the generic conceptual model from section 2.

In particular, to obtain the smallest number of competing solutions, make:

- *OMap* lowest in the ordering from definition 4
- $\vdash_{cov}$ highest in the ordering from definition 7
- $\vdash_{con}$ lowest in the ordering from definition 7
- Voc lowest in the ordering from theorem 9
- *Select* strongest in the ordering from theorem 10.

In other circumstances it might be desirable to obtain the *largest* number of competing diagnoses (for instance if we are concerned with not missing any possible diagnosis in a safety-critical situation). Similar validation results can be obtained for this purpose.

Besides validating the optimal number of competing diagnoses, we can also exploit the above and other similar theorems to validate other properties such as the size of individial diagnoses. In [tenTeije & vanHarmelen, 1996b] we have shown how we can also exploit the above and similar theorems to validate properties concerning the reliability and urgency of diagnoses.

6 Conclusions

In this paper we have taken a conceptual (= implementation independent) model of a wide class of diagnostic systems, and we have identified properties of this model that be used in both verification and validation. In section 3 we listed a number of properties that could be used to verify if a given conceptual model satifies minimum requirements of correctness. In section 4 we have given a number of theorems that can be used for two purposes: first to determine if a given conceptual model has certain as intended properties, and secondly to determine how the model should be adjusted if such property failed to hold.

This approach differs from much existing work on validation and verification of KBS in a number of ways:

Our conceptual model is phrased in task-specific ontology (observation, explanation, covering, etc.), in contrast with the task-independent ontology used in tradition work on V&V (premisse, conclusion, rule-set, etc.). This enables us to formulate a number of task-specific properties for the purposes of validation and verification (such as orderings of components that determine size and number of alternative diagnoses), in contrast with the task-independent properties traditionally investigated in V&V (loops in a rule-set, redundancy of a rule-set, etc.).

Besides being task-specific, our conceptual model (and consequently the properties based on it) are implementation-independent, again in contrast with traditional approach in V&V of KBS, which have usually focussed on properties of the implementation language (production rules, frames, etc.).

Two limitations of the work presented here are the following: first, we only deal with properties concerning the functionality of the KBS, while ignoring the dynamics of the reasoning process needed to realise this functionality. Our conceptual model is formulated using traditional predicate-logic (although a number of meta-layers are required, see footnote 2). In order to deal with the dynamics of the KBS, a richer formalism would be needed, such as the dynamic logic used by [Fensel *et al.*, 1996] or the temporal rules of [Cornelissen *et al.*, 1997].

A second limitation is that we only study properties concerning the inference steps of the KBS (our conceptual model corresponds to the inference layer of a KAD model, see section

2), and we have ignored the properties of the knowledge-base which required to support this reasoning (this would be the domain-layer, in KADS terminology). In a diagnostic context such properties would concern the presence of certain types of knowledge, the completeness of the knowledge, properties concerning the size and connectivity of the knowledge etc. Again, the work by [Fensel *et al.*, 1996] provides a good example of the kind of work that is needed in this direction.

References

[Benjamins, 1994] V. R. Benjamins. On a role of problem solving methods in knowledge acquisition – experiments with diagnostic strategies. In L. Steels, A. T. Schreiber, and W. van de Velde, editors, *LNAI 867*, EKAW'94, pages 137–157, Berlin, Germany, 1994. Springer-Verlag.

[Cornelissen *et al.*, 1997] F. Cornelissen, C. Jonker, and J. Treur. Compositional verification of knowledge-basaed systems: a case study for diagnostic reasoning. In *EUROVAV'97*, Leuven, June 1997.

[Fensel, 1995] D. Fensel. Specification languages in knowledge engineering and software engineering. *Knowledge Engineering Review*, Dec, 1995.

[Fensel *et al.*, 1996] D. Fensel, A. Schoenegge, R. Groenboom, and B. Wielinga. Specification and verification of knowledge-based systems. In *10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96)*, November 1996.

[Hacking, 1994] I. Hacking. What is logic? In D. Gabbay, editor, *What is a logical system?* Oxford Science Publications, 1994.

[Reiter, 1987] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–96, 1987.

[Schaerf & Cadoli, 1995] M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74(2):249–310, April 1995.

[tenTeije & vanHarmelen, 1996a] A. ten Teije and F. van Harmelen. Computing approximate diagnoses by using approximate entailment. In G. Aiello and J. Doyle, editors, *Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, Boston, November 1996. Morgan Kaufman.

[tenTeije & vanHarmelen, 1996b] A. ten Teije and F. van Harmelen. Examples of approximations in diagnosis based on approximate entailment. In S. Abu-Hakima, editor, *Seventh International Workshop on Principles of Diagnosis (DX'96)*, pages 239–248, Val Morin, Quebec, Canada, October 1996.

[tenTeije & vanHarmelen, 1996c] A. ten Teije and F. van Harmelen. Using reflection techniques for flexible problem solving. *Future Generation Computer Systems*, 12:217–234, 1996.

[tenTeije, 1997] A. ten Teije. *Automated Configuration of Problem Solving Methods in diagnosis*. PhD thesis, University of Amsterdam, March 1997.

[vanHarmelen & Aben, 1995] F. van Harmelen and M. Aben. Applying rule-base anomalies to KADS inference structures. In R. Gamble, editor, *IJCAI'95 Workshop on Validation and Verification of Knowledge-Based Systems*, Montreal, August 1995. to appear in *Decision Support Systems* journal, 1997.

[Wielinga *et al.*, 1992] B. J. Wielinga, A. Th. Schreiber, and J. A. Breuker. KADS: A modelling approach to knowledge engineering. *Knowledge Acquisition*, 4(1):5–53, 1992.