Chapter 16

Validation and refinement versus revision

F. Dupin de Saint-Cyr S. Loiseau
bannay@info.univ-angers.fr loiseau@info.univ-angers.fr
LERIA, Université d'Angers, 2 bd Lavoisier, 49045 ANGERS Cedex 01 France

Keywords: validation, refinement, coherency, diagnosis, revision, consistency handling, non monotonic inference relation

Abstract: This paper is a first work on the link between two fields: validation and refinement in knowledge base systems and revision in knowledge representation. The validation evaluates knowledge base quality; among the criterion which can define this quality an interesting one is coherency. The coherency of a set of rules can be viewed as the potential consistency of this set of rules with respect to every possible input. Refinement is a way to modify an incoherent knowledge base in order to restore its coherency. A revision operation is an inference relation, which is defined in order to reason under inconsistency, it provides a way to select rules to delete in order to restore the consistency. In the two fields, usually the initial knowledge base is consistent and the inconsistency can only come from some input on which a deduction must be done. This paper underlines the link between refinement and revision by comparing consistency to coherency and then by defining an inference relation for refinement.

1. INTRODUCTION

Numerous works have been done to propose criteria to measure the quality of a knowledge base (KB for short), and algorithms to evaluate those criteria. This kind of works can be grouped under the term *validation*. The validation process leads to proving either the respect of a given criterion, this provides an assessment about the KB quality, or the inadequacy of the KB with respect to this criterion. In this last case, the developer has to localise the cause(s) of this inadequacy, in order either to modify the rules of the KB such that it respects the criterion, or to assess that the causes of the inadequacy are not really revealing a bad quality in the KB but only a lack of integrity constraints. This process of localising causes of inadequacy, and finding potential modification can be called *refinement* process on a KBS. Validation and refinement have been specially studied in conjunction with real KBs, in order to provide tools to help KB developers. Many articles have been made on these topics in preceding EUROVAV conferences [Ayel95, VanHarmelen-VanVelde97]. In the validation field, information is divided into two parts: inputs and rules. In this paper, a KB is a set of rules. These rules give the meaning of the KB. A set of inputs is a set of facts. When the rules are applied to a set of inputs they leads to an output. For validation purpose, constraints are provided: they represent additional information, the constraints are settled to determine what inputs should never be dealt with or what output must be obtain from a given input. A particular constraint is a test constraint which is a mapping from an input to an output. Different criteria to measure the quality of a KB can be considered [Nguyen&al85]: *iredundancy, completeness, coherency.* Most research in validation has focused on KB coherence problems. A KB is coherent if and only if with any valid input, contradictory results cannot be inferred. The KB coherency can be defined as the consistency of the set of rules and constraints taken with each possible set of inputs. In other words, incoherency can be taken for potential inconsistency of the rules depending on inputs. The most difficult stage in this research is to transform this informal definition into formal and computationable (provable) criteria.

Apart from the validation field, a lot of researchers have studied inconsistency handling in KBs. Generally, the KB considered is a set of propositional logic formulas. Among all possible formulas, two particular cases can be distinguished: atomic formulas which represent ponctual facts, implicative formulas which represent general rules. Unlike in the validation field, in this kind of approach, facts and rules can be found together in the same KB. The KB available is then used in order to deduce new information about the system that it describes. The difficulty is to reason with an inconsistent KB, because the possible deductions become trivial, if we do not want to throw away the whole KB we have to deal with inconsistency. A particular problem is the insertion of a new fact (this fact being a new piece of information about the system represented) in an initially consistent KB; reasoning with the KB after the arrival of this new fact is called *revision* [Alchourrón&al85, Winslett88, Katsuno-Mendelzon91]. So, the problem of revision is to find which formula ψ can be deduced from a formula φ which has been added to the KB, in other words, to verify if φ added to the KB allows to infers ψ, but here the inference must not be the classical inference since φ can bring inconsistency to the KB and classical inference will allow to deduce anything and its contrary from an inconsistent KB. This is why, many researchers have proposed, so called, *non monotonic inference relations* which are able to deal with inconsistency.

This paper is a preliminary work to compare the two fields of research (validation versus revision) in order to see what techniques and theories may be exported from one to the other. This paper is organised as follows. In a first part, we are concerned with revision. Some examples of non monotonic inference relations are presented (this approaches are defined in [Benferhat&al93, DupindeSaintCyr&all94a,b]). In the second part, we present an approach to validate and refine rule bases. This approach was presented in [Bouali&al97], the validation is based on a rule base coherency criteria [Rousset88, Ginsberg-Williamson93], and the refinement uses the notion of diagnosis [Reiter87, Console&al93]. In the third part, we propose to compare the two approaches: (i) we compare consistency to coherency, then (ii) we show how to define an inference relation for refinement, this helps us to situate refinement with respect to revision.

2. KB REVISION

In the rest of this paper, we denote by $\mathcal{L}$ a finite propositional language. Elements of $\mathcal{L}$, or *formulas*, are denoted by Greek letters $\alpha, \beta,\ldots$ An *interpretation* in $\mathcal{L}$ is an assignment of a truth value in {T, F} to each formula of $\mathcal{L}$ in accordance with the classical rules of propositional calculus; we denote by Ω the set of all such interpretations. An interpretation ω is a *model* of a formula α ($\omega\models\alpha$) iff $\omega(\alpha)$=T. A formula β is called a logical consequence of α ($\alpha\models\beta$) iff each model of α is a model of β. A formula α which is satisfied by every interpretation is called a *tautology* (denoted by $\models\alpha$). A formula α is said to be consistent iff it has at least one model. Any inconsistent formula is denoted by $\perp$. Non monotonic inference relation will be denoted by $\mid\!\sim$. The problem of revision is to decide if given a knowledge base Δ and a new information φ, we can deduce ψ, denoted by $\varphi \mid\!\sim_{\Delta} \psi$. We recall that classical logic is *monotonic* in the sense that an inferred result is never questioned. In plausible reasoning such property is not always desirable since when we reason with general information we need to revise our conclusions in the light of new information. This is why we need to define non monotonic inference relations to reason with general information.

2.1 General definitions

In order to revise a knowledge base Δ by φ, many approaches have been proposed. The basic idea developed in the syntactical approaches (they are called syntactic because the way the formulas are written can influence the inferences: {p,q, ¬p∨¬q} will not behave like {p∧q, ¬p∨¬q}) is to select some consistent subsets of $\Delta \cup \varphi$, called preferred sub-theories, and to use the classical inference on these subsets.

Definition 1

If $\preceq$ is a preference relation on subsets of formulas the non monotonic inference relation based on $\preceq$ is defined by**:** $\varphi \mid\!\sim_{\preceq,\Delta} \psi \Leftrightarrow$ for all preferred (w.r.t. $\preceq$) sub-theory S of Δ consistent with φ, $S \cup \varphi \models \psi$

The point is to define a preference relation which is able to select the most interesting preferred sub-theories. In order to discriminate between the consistent subsets of KB, the selection criterion generally makes use of uncertainty considerations by using explicitly uncertainty measures [Wilson93, Benferhat&al95] or by using measures expressing priorities [Rescher64, Brewka89, Nebel91, Benferhat&al93, Lehmann92] (for a more detailed description of this kind of approach see [Cayrol-Lagasquie95]). These approaches consist in ranking the KB into priority levels (these levels are often supposed given by an expert (except in Pearl's system Z [Pearl90] where an automatic technique to rank the formulas is proposed, the algorithm is dedicated for KB in which formulas are default rules and the ranking is based on the specificity of the defaults) and maximising the set or the number of formulas satisfied at each level starting from the highest priority level. An important aspect of this kind of approach is that violating however many formulas at a given level is always more acceptable than violating only one formula at a strictly higher level: thus, these approaches are non-compensatory, i.e., levels never interact.

An alternative approach [Pinkas91, Dupin&all94a,b] is to weight the formulas of the KB with positive numbers called penalties. Intuitively, the penalty associated to a formula represents the importance of the formula, the higher it is, the more important is the formula and the more difficult it will be to reject this formula. Inviolable formulas are given an infinite penalty. Contrarily to priorities, penalties are compensatory since they are additive: the cost associated to a subset of formulas of a KB is the sum of the penalties of the rejected formulas, i.e., which are in the KB but not is this subset. The subset having a minimum cost are the preferred subset of KB. The penalties are supposed given by an expert.

2.2 Some examples of non monotonic inference relations

The first idea proposed in order to select among the possible sub-bases of an inconsistent KB, is to select sub-bases X1, ..., Xn which are consistent and maximal - either maximal for inclusion (X is a consistent sub-base maximal for inclusion if it does not exist a consistent sub-base Y such that X is included into Y) or maximal by its cardinality.

Example 1

Consider a domain where we want to determine if people can spent time as voluntary helpers. The variables WageEarner, Manager, Priest, HouseWife, are input variables which distinguish between different kinds of people. The variable SportHours denotes if the person can spent time (more than 2 hours) doing sport. FreeTime describes whether people have free time or not. HelpHours denotes if the person can spent time (more than 8 hours) as a volunteer helper. The following rules base RB describes our domain:

r1: WageEarner $\wedge$ Manager $\rightarrow$ $\neg$FreeTime
r2: $\neg$FreeTime $\rightarrow$ $\neg$HelpHours
r3: Priest $\rightarrow$ HelpHours
r4: HouseWife $\rightarrow$ HelpHours
r5: $\neg$FreeTime $\rightarrow$ $\neg$SportHours
r6: HouseWife $\wedge$ WageEarner $\rightarrow$ $\perp$

A question can be: what can we deduce from RB when we learn the facts WageEarner and Manager and Priest? It means that we look for the formulas ψ such that: WageEarner$\wedge$Manager$\wedge$Priest$\mid\!\sim_{\preceq,\ RB}\psi$. By definition of $\mid\!\sim_{\preceq,\ RB}$ it means that: $\forall$ S, S being a preferred (w.r.t. $\preceq$) sub-theory of RB consistent with WageEarner $\wedge$ Manager $\wedge$ Priest, S $\cup$ { WageEarner $\wedge$ Manager $\wedge$ Priest} $\models\psi$

If we use $\mid\!\sim_{\subseteq,\ RB}$: the preferred sub-bases of RB consistent with the formula WageEarner $\wedge$ Manager $\wedge$ Priest are: A={r1, r2, r4, r5, r6}, B={r1, r3, r4, r5, r6}, C={r2, r3, r4, r5, r6} (they are also preferred w.r.t. a cardinality ordering). The set of propositional variables that can be deduce from all the three bases is: {$\neg$HouseWife}. It means that WageEarner$\wedge$Manager $\wedge$Priest $\mid\!\sim_{\subseteq,\ RB}\neg$HouseWife. But we cannot decide if HelpHours is true or not with this inference relation.

Now, if the selection of the sub-bases is not strong enough, complementary information is needed about the KB, for so, the formulas of the KB can be weighted in order to represent a confidence degree which can be assessed to them by an expert. In this case, the KB can be stratified into priority levels (either directly by the expert or by an algorithm based on the syntax of the formulas like in Pearl’s system Z).

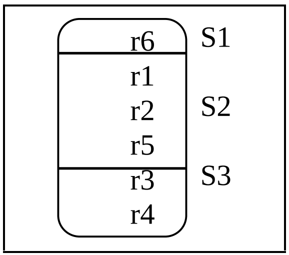

Figure 1. example of RB stratification into 3 priority levels

Now, the selection can take into account these levels, let S1, S2 , ... Sn be these levels ranked by decreasing priority. Let us present the lexicographic ordering (it is not the only way to order the sub-bases: you can also consider the best-out ordering proposed by [Dubois&all92] or the discrimin ordering proposed by [Brewka89], but it has been proved [Benferhat&al93] that A bo-preferred to B ⇒ A discrimin-preferred to B ⇒ A lex-preferred to B)

Definition 2 ([Benferhat&al93])

Let A and B be two consistent sub-sets of KB, A is *lexicographically-preferred* to B iff $\exists S_i$ such that $|B_i| < |A_i|$ and $\forall j < i, |A_j| = |B_j|$ (where $X_i = X \cap S_i$)

Example (followed)

A={r1,r2,r4,r5,r6}is the lexicographically-preferred sub-base consistente with WageEarner∧Manager∧Priest. It means that: WageEarner∧Manager ∧Priest $\mid\!\sim_{lex,RB}$ ¬HouseWife∧ ¬FreeTime∧ ¬HelpHours∧ ¬SportHours (since A ∪{WageEarner∧Manager∧Priest}⊨¬HouseWife∧¬FreeTime∧¬HelpHours∧¬Sport Hours .)

Instead of using priority levels, the formulas can be weighted with positive numbers called penalties. For any formula φ_i of the KB, there is an associated penalty α_i which represents a degree of confidence into φ_i, it will be understood as the cost that the user must pay in order to discard the formula φ_i. An infinite penalty is associated to unviolable formulas. The inference relation associated to a penalty KB use the following preference relation:

Definition 3 ([Dupin&all94a])

The cost C(A) of a sub-base A of KB is the sum of the penalties of the formulas in KB \ A. A sub-theory A of KB is *k-preferred* to a sub-theory B iff $C(A) \leq C(B)$.

Example 1

Let us consider the following penalty KB:

r1: WageEarner ∧ Manager → ¬FreeTime	$\alpha_1 = 20$
r2: ¬FreeTime → ¬HelpHours	$\alpha_2 = 20$
r3: Priest → HelpHours	$\alpha_3 = 5$
r4: HouseWife → HelpHours	$\alpha_4 = 5$

r5: $\neg$FreeTime $\rightarrow$ $\neg$SportHours $\alpha_5 = 20$
r6: HouseWife $\wedge$ WageEarner $\rightarrow \perp$ $\alpha_6 = +\infty$
The k-preferred sub-base of RB consistent with WageEarner$\wedge$Manager$\wedge$Priest is A (C(A)=5, C(B)=20, C(C)=20). So, WageEarner$\wedge$Manager$\wedge$Priest $\mid\!\sim_{k,RB}$ $\neg$HouseWife $\wedge\neg$FreeTime$\wedge\neg$HelpHours$\wedge\neg$SportHours.

Among the non monotonic inference relations which can be defined, some can be more appealing since they seem more “rational”. Kraus, Lehmann and Magidor [Kraus&al90] have proposed a set of properties for a non monotonic consequence relation. These properties are defined in order to capture the intended meaning of a “rational ” non monotonic relation. This set of properties is called System P and has been refined by adding a “rational monotony property” in [Alchourrón&al85].

3. KB VALIDATION-REFINEMENT

The validation approach is a bottom-up approach that attempts to measure the KB quality so that, if necessary, it can suggest to the expert to improve it. The KB refinement is supported by such a quality measurement. It aims to help the user to modify the KB in order to improve its quality. In rules base systems, many works [Rousset88, Beauvieux&Dagues90, Loiseau 92, Ginsberg-Williamson93, Bouali&al97] have been made on coherency.

In this part, rules are not exactly formulas of classical logic, a rule has the following form: "Ri: *If* L1 $\wedge$ ... $\wedge$ Ln *Then* Lm". Ri is the name of the rule. Li are literals or "$\perp$", we distinguish the subset, denoted by *R*, of rules of RB which have been stated as reliable by the designer and which cannot be modified. These rules are called *constraints*. Among these rules, those having the symbol "$\perp$" as a consequent are called *integrity constraints*. The other rules, denoted by *r* (RB = r $\cup$ R), are called *expert rules*. In the example, the name of the rules that belong to R begin with a capital letter. A fact is a literal (represented by a propositional variable or its negation). A fact base, denoted by *F*, is a set of facts. The *input facts* are the facts that can be given by a user during a session. An input fact is an instance of an *input literal*. The set of input propositional variables is determined once for all by the designer. We introduce the symbol $\vdash_{\mathcal{MP}}$ such that $\Delta \vdash_{\mathcal{MP}} \varphi$ if φ can be deduced by modus ponens from a set of facts and rules Δ (to a rule “*If* L1 $\wedge$ ... $\wedge$ Ln *Then* Lm” is associated the classical formula L1 $\wedge$ ... $\wedge$ Ln $\rightarrow$ Lm). The designer provides the rules base to validate as well as the input propositional variables set. To apply a rules base RB to a fact base F consists in computing the set of facts which can be deduced by modus ponens on F and RB.

In part 3.1, we present the RB VT_coherency property [Bouali&al97] which provides a measure to evaluate the quality of RBs. In part 3.2, we present a way to restore the VT_coherency of a RB, i.e., to refine the RB.

3.1 Validation

The VT_coherency property uses a set of test cases given by the designer. Formally, a test case is a pair <F_{test}, O> where F_{test} is an input fact base, and O (for Output) is a fact. Each test case is supposed to be certified as being reliable by the designer, it means that: $F_{test} \cup O \cup R \nvdash_{\mathcal{MP}} \perp$. In consequence, each test case can be represented by a logical association which is known to be held in the domain, between the input fact base F_{test} and the output fact O. This logical association is a constraint that can be associated with the test case.

A rules base is coherent if and only if with any valid input, contradictory results can not be inferred. Two main problems have to be solved. First, we have to know how to characterize a valid input. Second, we have to find a way to infer results from any valid input. In this approach, a valid input is an input fact base satisfying a set of constraints. These constraints are obtained from the rules base and the test cases.

Definition 4

Let <{f1, ..., fn}, O> be a test case, where {f1, ..., fn} is an input fact base, and O is a non input fact, " *If* f1 $\wedge$ ... $\wedge$ fn *Then* O" is the *associated constraint* to the test case, denoted by R_T.
Let T_{test} be a set of test cases, we call R_{test} the set of constraints associated with each test case of T_{test}.

Definition 5([Loiseau92])

An input fact base F_i *satisfies the constraints* R if $F_i \cup R \nvdash_{\mathcal{MP}} \perp$.

Example

r1: *If* WageEarner $\wedge$ Manager *Then* ¬FreeTime
r2: *If* ¬FreeTime *Then* ¬HelpHours
r3: *If* Priest *Then* HelpHours
r4: *If* HouseWife *Then* HelpHours
r5: *If* ¬FreeTime *Then* ¬SportHours
R6: *If* HouseWife $\wedge$ WageEarner *Then* $\perp$

Here, the last rule R6 is a constraint, the other rules are expert rules. The input literals set is {WageEarner, Manager, Priest, HouseWife, ManyChildren, ManyWorkHours}. Consider that we have the following set of test cases:
T1:<{WageEarner, Manager, ManyWorkHours}, ¬HelpHours>
T2:<{HouseWife, ManyChildren}, ¬HelpHours>
T3:<{Priest}, HelpHours>
T4:<{WageEarner, Manager, ¬ManyWorkHours}, SportHours>
T5:<{WageEarner, Manager, ManyWorkHours}, ¬SportHours>
R_{test} is composed of 5 constraints associated to T1,...,T5. For example, the constraint associated with T2 is: "*If* HouseWife$\wedge$ManyChildren *Then* ¬HelpHours". F={House

Wife,ManyChildren,Priest} does not satisfy the constraints $R \cup R_{test}$ (since $F \cup \{R_{T2}, R_{T3}\} \vdash_{\mathcal{MP}} \perp$)

Definition 6 ([Bouali&al97])

A rules base RB is *VT_coherent* with respect to a set of test cases T_{test} if
1. For each input fact base F_i satisfying the constraints $R \cup R_{test}$, $F_i \cup RB \nvdash_{\mathcal{MP}} \perp$
2. For each test case $<F_{test}, O> \in T_{test}$, $F_{test} \cup O \cup RB \nvdash_{\mathcal{MP}} \perp$.

The VT_coherency property generalizes and improves the properties given in the following works: [Rousset88] [Ginsberg88b] [Beauvieux-Dagues90] [Loiseau92] [Ginsberg-Williamson93] and [Levy-Rousset96].

Example

The KB presented above is VT_incoherent. On one hand, {WageEarner, Manager, Priest} is an input fact base satisfying $R \cup R_{test}$ ({WageEarner, Manager, Priest} $\cup$ R6 $\cup$ $R_{test} \nvdash_{\mathcal{MP}} \perp$), and on the other hand, it enables the firing of r1, r2 and r3 leading to the contradiction: HelpHours and ¬HelpHours. Hence, the VT_incoherency.
We can remark that with T2 a contradiction can be obtained, because the condition part of T2 {HouseWife, ManyChildren} provides a way to fire r4 whose conclusion is " HelpHours " which is contradictory with the conclusion of the test case T2: " ¬HelpHours ". This is an other way to show the VT_incoherency. In some cases, it is only the test cases that bring incoherencies.

3.2 Refinement

Once a rule base is VT_incoherent, we have to refine it to restore the coherency. On one way, we can define a minimal proof of an incoherency as a conflict composed of an input and a set of rules. On the other way, many conflicts can exist, so we define a cause of all the incoherences as a diagnosis which is composed of inputs and rules such that if those inputs and those rules are forbidden then the rest of the rule base is VT_coherent. This process to compute diagnoses is called the refinement process because it provides a way to restore the coherency by modifying the rules base. We do not present conflicts in this paper, cf. [Bouali&al97] for more details.

By analogy with the debugging of a classical program, providing a conflict should be equivalent to providing an input along with the appropriate part of the program leading to the possible invalidity. It should be more efficient to provide the designer with a possible cause of the VT_incoherency (a diagnosis) of the rules base rather than a proof of the VT_incoherency (a conflict). Continuing the analogy with the debugging of a classical program, a cause should be one or a few lines of code whose incorrectness could be sufficient to explain the invalidity of the program (all the conflicts).

The VT_incoherency of a rules base can be explained (i) by the lack of constraints which could exclude the input fact bases leading to a contradiction because they are not valid in the domain, and/or (ii) by bad rules that lead to a contradiction with valid input fact bases. A possible cause of the VT_incoherency is called diagnosis. A rules base diagnosis is composed by a set of input fact bases and a set of expert rules. To each input fact base $F_i = \{f_{i,1}, ..., f_{i,n}\}$ of a diagnosis can be associated an integrity constraint " *If* $\wedge_j f_{i,j}$ *Then* $\perp$ ". Note that the more numerous the facts are in the integrity constraint and the less restrictive is this constraint, this is why, all the facts of F_i are taken in the conjunction of the integrity constraint.

Definition 7

Let us consider a pair D=< E_D, r_D>, E_D being a set of input fact bases $E_D = \{\{f_{1,1}; ..., f_{1,n}\}; ...;\{f_{p,1}; ...; f_{p,m}\}\}$ and r_D beeing a subset of r (the expert rules). Let us call R_{ED} the set of the p integrity constraints " *If* $\wedge_j f_{i,j}$ *Then* $\perp$ " associated to E_D. Let D(RB) be the rules base repaired by the pair < E_D, r_D>, it corresponds to RB from which the rules r_D has been deleted and the p integrity constraints of R_{ED} are added: D(RB) = $(R \cup r \setminus\{r_D\}) \cup R_{ED}$.
< E_D, r_D> is a *diagnosis* of an incoherent rules base RB, with respect to a set of test cases T_{test}, if D(RB) is VT_coherent with respect to T_{test}.

A VT_incoherent rules base becomes VT_coherent if some integrity constraints associated to the fact bases of the diagnosis are added and if the rules of the same diagnosis are deleted.

Definition 8

A diagnosis <E_D, r_D> is *minimal* if there does not exist another diagnosis <E_D', r_D'> verifying: $r_D' \subseteq r_D$, and ($E_D' \subseteq E_D$, or $\forall$ F' in E_D', $\exists$ F in E_D such that $F \subseteq F'$).

Examples

- Considering the two diagnosis E1= <{{a,b},{a, c}}, {r1}> and E2=<{{a,b}}, {r1}>, E2 is minimal, since it is sufficient to forbid the conjunction of the litterals a and b and to suppress the rule r1 in order to apply the diagnosis E2, meanwile two conjunctions are forbidden in E1.
- Considering the diagnosis E3=<{{a }}, {r1}>, E1 is minimal in the set of diagnosis {E1,E3} since having a constraint which forbids a is more restrictive that having two constraints forbidding $a \wedge b$ and $a \wedge c$.

Our approach considers only minimal diagnoses that contain a set of bad rules and missing constraints sufficient to restore the VT_coherency. Of course, several minimal diagnoses often exist. The choice of the right diagnosis to repair the rules base can be done using heuristics and/or with the help of the designer of the rules base. Here we propose to select the diagnosis which are not only consistent with the set of test cases but also do not block the derivation of the expected output from an input which is given in a test case, see [Bouali&al97] for more details.

Definition 9

$D=<E_D,r_D>$ is a *prefered* diagnosis of a rules base RB with respect to a set of test cases T_{test}, if D is a minimal diagnosis and for each test case $<F_{test}, O>$ in T_{test} s.t. $F_{test} \cup RB \vdash_{\mathcal{MP}} O$, the rules base repaired by D, D(RB), verifies $F_{test} \cup D(RB) \vdash_{\mathcal{MP}} O$.

The algorithm to compute the “diagnoses” of a physical system [Reiter87] in terms of “conflicts” proving the incoherency of the system can be extended to compute the minimal diagnoses of a rules base when the conflicts of the rules base are known [Bouali&al97].

4. DISCUSSION

Two main differences can be underlined between validation-refinement and revision. First, validation uses a coherency property that is not used by revision which uses only classical consistency. Second, revision proposes many formal ways to modify a KB, meanwhile refinement proposes more informal solutions to modify a KB.

First, let us compare consistency and coherency. To each rule of a rules base a logical implication can be naturally associated. Under our assumptions, the functioning of an inference engine can be assimilated to that of a logical demonstrator based on modus ponens. It seams natural to evaluate the rules base coherency by testing the consistency of the logical formulae associated with the rules base. Numerous algorithms have been studied for this and could be used to evaluate the logical consistency of the set of rules. However, the existence of a model of the set of logical implications associated to the rules is a condition too weak to evaluate the coherency of the rules base.

Example 1

Let us come back again to our example expressed in classical logic:

r1: WageEarner ∧ Manager → ¬FreeTime
r2: ¬FreeTime → ¬HelpHours
r3: Priest → HelpHours
r4: HouseWife → HelpHours
r5: ¬FreeTime → ¬SportHours
r6: HouseWife ∧ WageEarner → ⊥

This base is consistent; for instance, the interpretation {WageEarner, Manager, ¬FreeTime, ¬HelpHours, ¬Priest, ¬HouseWife, ¬SportHours} satisfies it. But if we consider that {WageEarner, Manager, Priest} is a valid input in the domain, we would like to say that RB is not “coherent ” according to our coherency property.

{WageEarner, Manager, Priest} provides (with r1, r2, r3) {¬HelpHours} and {HelpHours} which are contradictory literals.

In the validation field, the coherency of a rules base depends on the input fact bases that a user could give during the utilisation phase: we call such inputs *valid inputs*. Each valid input can be translated into the conjunction of its facts. What coherency requires is that each formula associated to a valid input taken with the set of formulas associated with the rules base can not infer the contradiction by modus ponens.

Some difficulties appear. First, the cost of such an approach is proportional to the number of valid inputs. Second, the number of valid inputs can be infinite. Third, the way to obtain such valid inputs or a modelisation of them must be found: as soon as we do not consider given test cases as the only valid inputs, the difficulties arise, there is no efficient and simple solution. So, even if works on classical logic can provide some results to researchers who are studying the problem of the evaluation of the coherency of a rules base, they are not sufficient to provide a correct and complete solution to the coherency problem as we have presented it. This is why we introduce (VT) coherency definitions.

Second, is refinement by diagnosis a kind of revision? As we said before, computing the potential consistency of a rule base cannot be done reasonably because the potential inputs of the rules base can be very numerous and even infinite. So, for consistency detection the revision field can not bring any help to the validation field. Let us see if the refinement proposed above can be viewed as a revision operator.

Definition 10

Given a rules base RB and a set of test cases T, let us define the relation $\mid\!\sim_{diag}$ as: $\varphi \mid\!\sim_{diag, RB, T} \psi \Leftrightarrow \varphi$ is a conjunction of input facts, and ψ is a conjunction of output facts s.t.: $\varphi \cup R \cup R_{test} \nvdash_{\mathcal{MP}}$ (R_{test} being the set of constraints associated to T) and

a) if RB is VT-coherent then $\varphi \cup R \cup r \vdash_{\mathcal{MP}} \psi$

b) else for each prefered diagnosis $<E_D,r_D>$:

$$\varphi \cup R \cup R_{test} \cup R_{ED} \nvdash_{\mathcal{MP}} \text{ and } \varphi \cup R \cup r \backslash \{r_D\} \vdash_{\mathcal{MP}} \psi.$$

In other words, if RB is VT-coherent with T, from φ we infer ψ, if and only if φ is a valid input and taken with the RB, it allows to infer ψ by modus ponens. In the other case, i.e., if RB is VT-incoherent, for any prefered diagnosis $<E_D, r_D>$, φ must be valid with respect to the diagnosis E_D (i.e., with R_{ED}) and taken with RB from which the rules r_D are removed it should infer ψ by modus ponens.

Here, the inference relation is defined under particular formulas, the formulas φ and ψ must be under the form of a conjunction of facts, it is impossible to infer an implication for instance, or to put a rule as an input. Moreover, the inference can only take into account valid inputs.

Besides, the computation of diagnoses can be done once for all and then the inference relation uses an inference by modus ponens based upon a new rules base. This computation does not depend upon the input.

The inference relation defined is a very particular revision operator, since the KB can be modified once for all with respect to the set of test cases and then from any valid input an output can be computed. In general, a classical revision operator must take into account each input in order to update the knowledge contained into the KB and then the inference can be done.

In conclusion, though *revision* can not bring any help in the field of *validation* because there is no algorithms allowing to compute potential inconsistancies or, as we call them, incoherencies, revision can be linked with *refinement*. We have seen that an inference relation can be defined from a rules base RB and a set of test cases T by the computation of diagnoses. A diagnosis consists in a set of rules to delete and a set of conjunctions of facts to forbid. The inference relation we propose is an inference by modus ponens which takes into account a new rules base RB' computed once for all from RB and T (if RB is VT-coherent with T, RB'=RB else RB' is RB from which some rules are deleted and some integrity constraints are added following a choosen diagnosis). In fact, in case of VT-incoherency, there are many ways to choose a diagnosis which can repair RB into an RB'. In this paper a preference relation on diagnosis is proposed. The inference relation must then examin every possible rules base RB' corresponding to each preferred diagnosis.

An interesting perspective of this article, is to use revision techniques in order to refine the selection of preferred diagnoses: this selection could take into account some preferences on the rules as it is shown in the first part of this paper.

References

[Alchourrón&al85] Alchourrón, Gärdenfors and Makinson. On the logic of theory change: partial meet contraction and revision functions. Jl of Symbolic Logic, vol 50, 510-530, 85.

[Ayel95] EUROVAV (European Symposium on the Validation and Verification of Knowledge Based-Systems), Chambéry, 1995.

[Beauvieux-Dagues90] Beauvieux, A., Dague, P. A general consistency (checking and restoring) engine for Knowledge Bases. ECAI, p.77-82 (1990).

[Benferhat&al93] Benferhat, Cayrol, Dubois, Lang, Prade. Inconsistency management and prioritized syntax-based entailment. IJCAI, vol 3, 640-645. 1993.

[Benferhat&all95] Benferhat and Smets. Belief functions for logical problems: representing default rules in ε-beliefs logics. In Abstract of the Dagstuhl Seminar, 1993.

[Bouali&al97] Bouali, Loiseau, Rousset, M.C. Revision of Rule Bases. EUROVAV 1997.

[Brewka89] G. Brewka. Preferred subtheories: an extended logical framework for default reasoning. IJCAI, 1043-1048. 1989.

[Cayrol-Lagasquie95]. Cayrol, Lagasquie-Schiex. Non-monotonic syntax-based entailment: a classification of consequence relations. Lecture notes in AI, 946, p. 107-114.1995.

[Console&al 93] Console L., Friedrich G., Dupré D.T. Model-Based Diagnosis Meets Error Diagnosis in Logic Programs, IJCAI, p.1494-1499 (1993).

[Dubois&al92] Dubois, Lang, Prade. Inconsistency in possibilistic knowledge bases -To live or not live with it. Fuzzy Logic for the Management of Uncertainty, Wiley, 335-351.1992.

[Dupin de Saint-Cyr&all94a] Dupin de Saint-Cyr, Lang and Schiex. Gestion de l'inconsistance dans les BC: une approche syntaxique basée sur la logique des pénalités.RFIA,94.

[Dupin de Saint-Cyr&all94b] Dupin de Saint-Cyr, Lang and Schiex. Penalty logic and its link with Dempster-Shafer theory. In Proc. of the 10^{th} Uncertainty in AI, p. 204-211. 1994.

[Ginsberg-Williamson93] Ginsberg, Williamson, Inconsistency and Redundancy Checking for Quasi-First-Order-Logic Knowledge Bases. International Journal of Expert Systems, vol.6(3), p.321-340 (1993).

[Katsuno-Mendelzon91] Katsuno, Mendelzon. On the difference between updating a knowledge base and revising it. Proc. of the 2nd Inter. Conf. on Principles of Knowledge Representation and Reasoning, p. 387-394, Cambridge, MA, 1991.

[Kraus&al90], Kraus S., Lehmann D., Magidor M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence, 44, 167-207.

[Lehmann92] D. Lehmann. 1992. Another perspective on default reasoning. Tec. Report.

[Levy-Rousset96] Levy, A. Rousset, M.C. Verification of knowledge bases based on containment checking. AAAI, p.585-591 (1996).

[Loiseau92] Loiseau. Refinement of Knowledge Bases Based on Consistency. ECAI, p. 845-849 (1992).

[Nebel91] B. Nebel. Belief revision and default reasoning: syntax-based approaches. In Proc. of the 2nd KR, p. 417-428. Cambridge, MA, 1991.

[Nguyen&al85] Nguyen, T.A., Perkins, W.A., Laffey, T.J., Pecora, D. Checking an expert systems knowledge base for consistency and completeness. IJCAI, p.375-379 (1985).

[Pearl90] J. Pearl. System Z: A natural ordering of defaults with tractable applications to default reasoning. Proc. of Theoretical Aspects of Reasoning about Knowledge, pages 121-135. Morgan Kaufman, San Mateo. 1990

[Pinkas91] G. Pinkas. Propositional nonmonotonic reasoning and inconsistency in symmetric neural networks. In Prof. of the 12th IJCAI. p. 525-530. Sydney, Australia, 1991.

[Reiter87] Reiter, R. A theory of diagnosis from first priciples. AI, vol 32, p. 57-95 (1987).

[Rescher64] N. Rescher. Hypothetical Reasoning. North-Holland, 1964.

[Rousset88] Rousset, M.C. On the consistency of knowledge bases: the COVADIS system. ECAI, p.79-84 (1988).

[VanHarmelen-VanVelde97] EUROVAV (European Symposium on the Validation and Verification of Knowledge Based-Systems) 1995, Leuven.

[Wilson93] N. Wilson. Default logic and Dempster-Shafer theory. ECSQARU, p. 372-379, Granada, Spain 1993.

[Winslett88] M. Winslett. Reasoning about action using a possible models approach. Proc of the 7th National Conference on Artificial Intelligence, p. 89-93, St. Paul, 1988.