Handling Conflicts in Access Control Models

Salem BENFERHAT and **Rania EL BAIDA**[1]

1 INTRODUCTION

The problem of handling conflicts is central in many information processing areas. In information security it often happens that security requirements are conflicting. For instance, a confidentiality requirement may specify that a given user u is prohibited to have an access to some sensitive data. But, an availability requirement may specify that u is obliged to have an access to these data in order to perform some urgent and critical action.

There have been several proposals for handling conflicts in propositional knowledge bases [2]. Most of them use stratified knowledge bases of the form $\Sigma = S_1 \cup \ldots \cup S_n$, such that formulas in S_i have the same level of priority and have higher priority than the ones in S_j where $j < i$.

The following example shows that the "blind" application of coherence-based propositional approaches, like "cardinality inference" [2] , to inconsistent first order knowledge bases can lead to undesirable conclusions.

Example 1 *Assume that we have the following rules and facts:*
$\mathbf{R_1}.\forall x \forall y, Cardio(x,y) \rightarrow PermRead(x,y)$ *(cardiologists have permission to read their patients' surgical records).*
$\mathbf{R_2}.\forall x \forall y, Cardio(x,y) \wedge onstrike(x) \rightarrow \neg PermRead(x,y)$ *(cardiologists, who are on strike, do not have the permission to read patients' surgical records).*
$\mathbf{R_3}.\forall x \forall y, Phys(x,y) \rightarrow \neg PermRead(x,y)$ *(physicians do not have permission to read the patients' surgical records).*
$\mathbf{F_1}.Cardio(John, JO)$ *(JO is a patient of the cardiologist John).*
$\mathbf{F_2}.Phys(John, JO)$ *(JO is a patient of the physician John).*
$\mathbf{F_3}.onstrike(John)$ *(John is on strike).*
$\mathbf{F_4}.Cardio(Bob, JO)$ *(JO is a patient of the cardiologist Bob).*
$\mathbf{F_5}.Phys(Bob, JO)$ *(JO is a patient of the physician Bob).*
$\mathbf{F_6}.\neg onstrike(Bob)$ *(Bob is not on strike).*
Assume that these rules are ranked in the following way: $\Sigma = S_1 \cup S_2 \cup S_3 \cup S_4$*, with:*
$S_4 = \{F_1, F_2, F_3, F_4, F_5, F_6\}$; $S_3 = \{R_2\}$;
$S_2 = \{R_1\}$; $S_1 = \{R_3\}$.
Intuitively, we expect to conclude that Bob *is permitted to read JO's surgical record (since he is not on strike and he is one of the JO's cardiologists).*
From Σ*, the only cardinality based preferred subbase is:* $A = S_4 \cup S_3 \cup S_1$*, from which the undesirable conclusion,* $\neg PermRead(Bob, JO)$*, is deduced.*

This "adventurous conclusion" is explained by the fact that cardinality-based systems proceed level by level, and after removing $\mathbf{R_1}$ (since it is conflicting with the fact that $John$ is on strike), $\mathbf{R_3}$ can now be applied for Bob.

[1] CRIL,Université d'Artois, Rue Jean Souvraz, SP18 F62307 Lens, France email: {benferhat,elbaida}@cril.univ-artois.fr

2 FIRST-ORDER COHERENCE-BASED APPROACHES

The limitations of coherence-based approaches are explained by the fact that removing one first order formula comes down to the removing of a set of all its instantiated propositional formulas. This is not satisfactory since if a formula is responsible of a conflict, then it is not the case that all its instances are also responsible of a conflict. In this section, we propose to appropriately redefine the cardinality-based inference [2] in first order logic framework. The idea is that a formula of the form $\forall x \phi(x)$ responsible of a conflict should not be deleted, except if all its instances are also responsible of a conflict. We propose to weak this formula, namely to drop only instances of this formula which are responsible of a conflict. For instance, if a formula of the form $\forall x\ \phi(x)$ is conflicting for $x = a$, then this formula will be replaced by $\forall x, \neg(x = a) \rightarrow \phi(x)$.

Let ϕ be a formula referring to an uncertain rule, which is universally quantified with a set of variables $X = \{x_1, \ldots, x_n\}$. Let $I = \{i_1, \ldots, i_n\}$ be such that i_k's are instances of x_k's respectively. We use the predicate $Diff(I, X)$ to represent the formula $\neg(\bigwedge_{k=1,\ldots,n}(x_k = i_k))$.

Definition 1 *Let* ϕ *be an uncertain formula.* ϕ_{weak} *is called a weakened formula of* ϕ *if it has the form:*
$A \Rightarrow \phi$*, where* $A = \{Diff(I_j, X) : j = 1, \ldots, n\}$.
We define the degree of a weakened formula, ϕ_{weak}*, as simply equal to the cardinality of A, namely* $degree(\phi_{weak}) = |A|$.

Intuitively, the degree of a weakened formula represents the number of instances that cannot be applied (i.e., are ignored).

The counterpart of consistent subbases in the case of propositional inconsistent knowledge bases is the notion of weakened first order knowledge bases in the first order framework. More formally,

Definition 2 *A first-order knowledge base* $\Sigma' = S'_1 \cup \ldots \cup S'_n$ *is said to be a weakened base of* $\Sigma = S_1 \cup \ldots \cup S_n$ *if i)* Σ' *is consistent, and ii)* Σ' *is only obtained by replacing some uncertain formulas* ϕ *of* $S_1 \cup \ldots \cup S_{n-1}$ *by their weakened counterpart* ϕ_{weak}.

The following definitions introduce the degree associated with a stratum of a weakened base and the preference relation between weakened bases:

Definition 3 *We define the degree of a stratum* S'_i*, of a weakened base* Σ'*, as:* $degree(S'_i) = \sum_{\phi_{weak} \in S'_i} degree(\phi_{weak})$.

Definition 4 *Let* Σ' *and* Σ'' *be two weakened bases of* Σ*.* Σ' *is said to be 1-preferred to* Σ''*, denoted by* $\Sigma' >_1 \Sigma''$*, if* $\exists i, 1 \leq i \leq n$ *with*
i) $degree(S'_i) < degree(S''_i)$*, and*
ii) $\forall j > i, degree(S'_j) = degree(S''_j)$.
Σ' *is said to be 1-preferred weakened base of* Σ *if there is no consistent weakened base* Σ'' *such that* $\Sigma'' >_1 \Sigma'$.
Lastly, a formula ψ *is a 1-Card conclusion of* Σ*, denoted* $\Sigma \vdash_1 \psi$*, if* ψ *is a consequence of all 1-preferred weakened bases of* Σ.

Example 2 *Let us consider again Example 1. Let Σ' be a weakened base of Σ such that $\Sigma' = S'_1 \cup S'_2 \cup S'_3 \cup S'_4$ with $S'_4 = S_4$, $S'_3 = S_3$, S'_2 is obtained by replacing the formula $\mathbf{R_1}$ of S_2 by*
$\phi_{weak} = \forall x \forall y, Diff(\{John, JO\}, \{x, y\}) \wedge Cardio(x, y) \rightarrow PermRead(x, y)$,
S'_1 is obtained by replacing the formula $\mathbf{R_3}$ of S_1 by
$\phi_{weak} = \forall x \forall y, Diff(\{Bob, JO\}, \{x, y\}) \wedge Phys(x, y) \rightarrow \neg PermRead(x, y)$.
It can be checked that Σ' is the only 1-preferred base of Σ. From this new base Σ', we can deduce that $\Sigma' \vdash PermRead(Bob, JO)$, as expected.

3 APPLICATION TO OrBAC ACCESS CONTROL SYSTEMS

This section applies handling conflicts to a recent access control model, called OrBAC (Organization-Based Access Control System) [1]. Figure 1 illustrates OrBAC System. The idea is that the privileges attribution is not explicitly made for each user, object and action but rather to their abstraction entities *Role, View* and *Activity*, with respect to a given organization.

A security policy is a set of permissions (resp. prohibitions) rules which are defined using the relationships *Permission* and *Prohibition*. The relationship $Permission(org, r, a, v, c)$ means that the organization org grants the role r a permission to perform an activity a on a view v within the context c. The relationship $Prohibition(org, r, a, v, c)$ is defined similarly.

Contexts specify the circumstances where organizations grant roles permissions to perform activities on views. They are defined using the relationship $DefContext(org, s, \alpha, o, c)$.

Access control must provide a framework for describing the concrete actions that may be performed by subjects on concrete objects. For this purpose, the relationships $Is\_permitted(s, \alpha, o)$ and $Is\_prohibited(s, \alpha, o)$ are introduced.

We use the relationships *Employ(org,s,r)*, *Use(org,o,v)* and *Consider(org,α,a)* to jump from abstract to concrete privileges, as it is illustrated in Figure 1.

We denote the jumping rules (from abstract to concrete permission) $\phi_P(org, r, a, v, c, s, o, \alpha)$ of the form: $\forall org \forall s \forall \alpha \forall o$
$Permission(org, r, a, v, c) \wedge Employ(org, s, r) \wedge$
$Use(org, o, v) \wedge Consider(org, \alpha, a) \wedge$
$DefContext(org, s, \alpha, o, c) \rightarrow Is\_permitted(s, \alpha, o)$.
$\phi_I(org, r, a, v, c, s, o, \alpha)$ is defined similarly for prohibition.
ϕ_P and ϕ_I are the only uncertain rules.

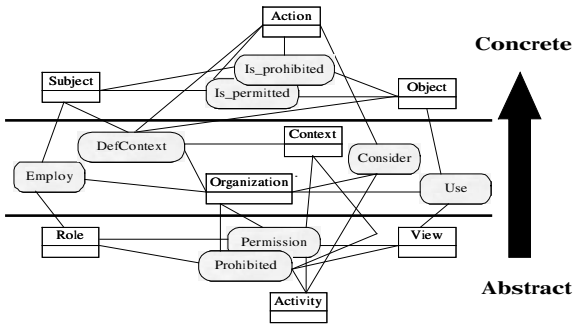

Figure 1. The OrBAC model

Example 3 *Let us consider again example 1, and let us suppose that these rules are associated within the organization H. We extract two roles "cardio" and "phys", and three contexts "patient", "strike" and "normal". The encoding of these rules and facts is as follows (universal quantifications are omitted for sake of simplicity).*

Set of facts:
$\mathbf{F_1}$. $Permission(H, cardio, consult, record, patient)$.
$\mathbf{F_2}$. $Prohibition(H, cardio, consult, record, strike)$.
$\mathbf{F_3}$. $Prohibition(H, phys, consult, record, normal)$.
$\mathbf{F_4}$. $Consider(H, read, consult)$.
$\mathbf{F_5}$. $Use(H, recJO, record)$.
$\mathbf{F_6}$. $Employ(H, John, cardio)$. $\mathbf{F_7}$. $Employ(H, John, phys)$.
$\mathbf{F_8}$. $Patient(JO, John)$. $\mathbf{F_9}$. $OnStrike(John)$.
$\mathbf{F_{10}}$. $Employ(H, Bob, cardio)$. $\mathbf{F_{11}}$. $Employ(H, Bob, phys)$.
$\mathbf{F_{12}}$. $Patient(JO, Bob)$.
Set of completely certain rules:
$\mathbf{R_1}$. $\neg Is\_permitted(s, \alpha, o) \vee \neg Is\_prohibited(s, \alpha, o)$.
$\mathbf{R_2}$. $DefContext(H, s, read, recJO, patient) \leftrightarrow Patient(JO, s)$.
$\mathbf{R_3}$. $DefContext(H, s, read, recJO, strike) \leftrightarrow OnStrike(s)$.
$\mathbf{R_4}$. $DefContext(H, s, read, recJO, normal) \leftrightarrow \top$.
Set of uncertain rules:
$\mathbf{URP_1}$. $\phi_P(org, r, a, v, patient, s, o, \alpha)$.
$\mathbf{URP_2}$. $\phi_P(org, r, a, v, strike, s, o, \alpha)$.
$\mathbf{URP_3}$. $\phi_P(org, r, a, v, normal, s, o, \alpha)$.
$\mathbf{URI_1}$. $\phi_I(org, r, a, v, patient, s, o, \alpha)$.
$\mathbf{URI_2}$. $\phi_I(org, r, a, v, strike, s, o, \alpha)$.
$\mathbf{URI_3}$. $\phi_I(org, r, a, v, normal, s, o, \alpha)$.

Assume that the stratified knowledge base is:
$\Sigma = S_4 \cup S_3 \cup S_2 \cup S_1$, *with* $S_4 = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, R_1, R_2, R_3, R_4\}$, $S_3 = \{URI_2\}$, $S_2 = \{URP_1, URP_2, URP_3\}$, *and* $S_1 = \{URI_1, URI_3\}$.
Σ *is inconsistent. Let Σ' be the weakened base of Σ defined by:* $\Sigma' = S'_1 \cup S'_2 \cup S'_3 \cup S'_4$, *with* $S'_4 = S_4$, $S'_3 = S_3$, $S'_2 = \{URP'_1, URP_2, URP_3\}$, *and* $S'_1 = \{URI_1, URI'_3\}$, *where*
S'_1 *is obtained by replacing* $\mathbf{URI_3}$ *of* S_1 *by:*
$\mathbf{URI'_3}$. $\phi_{WeakI}(org, r, a, v, normal, s, o, \alpha) \equiv$
$Diff(\{Bob, read, recJO\}, \{s, \alpha, o\}) \rightarrow$
$\phi_I(org, r, a, v, normal, s, o, \alpha)$,
and S'_2 is obtained by replacing $\mathbf{URP_1}$ *of* S_2 *by:*
$\mathbf{URP'_1}$. $\phi_{WeakP}(org, r, a, v, patient, s, o, \alpha) \equiv$
$Diff(\{John, read, recJO\}, \{s, \alpha, o\}) \rightarrow$
$\phi_P(org, r, a, v, patient, s, o, \alpha)$.
Σ' *is consistent and is the only 1-preferred base of Σ. We can deduce that $\Sigma' \vdash Is\_permitted(Bob, read, recJO)$, as expected.*

4 CONCLUSION

This paper showed that the blind application of propositional approaches, to inconsistent first order knowledge bases, is not satisfactory. We proposed a solution based on weakening a formula rather than removing it. We showed how this can be used to manage inconsistencies in OrBAC system.

Acknowledgment This work is supported by the national ACI project DESIRS.

REFERENCES

[1] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel, and G. Trouessin. Organization based access control. In *Policy'03*, pages 120–131, 2003.

[2] S. Benferhat, D. Dubois, and H. Prade. How to infer from inconsistent beliefs without revising? In *IJCAI'95*, pages 1449–1455, 1995.