Towards handling dynamic security policies

Rouza Ait Tahar

Salem Benferhat
CRIL-CNRS, LENS

Abstract: Establishing formal models of security policies to control and restrict the access to sensitive pieces of information is a very challenging problem with applications in many domains. Several formalisms have been proposed for the modeling of security policies. However, these formalisms are based on static access control policies and do not take into account the evolution of systems and of the security policies required of them. This paper proposes an approach for the revision of security policies. Policy revision refers here to addition or suppression of regulations.

1 Introduction

Information systems more and more manage sensitive and critical data. Therefore, it is very important to define security policies to restrict access to these data in order to guarantee security properties.
Several works have been proposed for modeling access control rules in an information system. Most of these models are based on the concept of roles associated with users as in the RBAC (*Role-Based Access Control*) system. In these systems, access rights are assigned to users according to their roles played in the system.
However, these systems are based on static access control policies and do not take into account the evolution of systems and of the security policies required of them. For example, it is impossible to revise a security policy by the imposition of new regulations (that should usually take precedence over previous ones). Revising security policies is unavoidable in practice, be it in order to integrate new regulations or in order to react to and correct security faults that have been detected.
Our objective relates to the revision of security policies. Policy revision can refer to addition or suppression of regulations. Introducing new regulations poses a problem when the new rules contradict or question existing regulations. We envision techniques to identify a minimal set of rules that need to be reconsidered in order to maintain a coherent policy. Dually, the suppression of existing rules can be problematic that requires to inspect the intention of the modi cation. For example, it is of no use to remove some rule granting permission to access a piece of information when this permission can be obtained in a different way: a single modification may affect a number of regulations, and rules should again be managed to ensure the coherence of the overall policy.
Our solution is developped within possibilistic logic framework. Possibilistic logic offers a convenient tool for handling uncertain or prioritized formulas and coping with inconsistency. Propositional logic formulas are thus associated with weights belonging to a linearly ordered scale.

Sections 2 and 3 present basic concepts used in the OrBAC model and its formalization in a first order logic framework [Aboulkalem et al2003]. Section 4 shows how to revision is handled in the possibilistic logic framework.

2 Organization-based access control

This section briefly presents basic concepts of the OrBAC model [Aboulkalem et al2003], using a diagrammatic language based on the entity-relationship model.

2.1 Basic ”concrete” concepts The basic concepts are organizations (the most important entity in the OrBAC model), subjects, objects and actions. An *Organization* can be seen as an organized group of subjects playing some roles. A *Subject* is basically a user (i.e., a human being, e.g., John). The entity *Object* covers inactive entities (e.g., system’s files). The entity *Action* contains computer actions such as ”read”, ”write”, etc.

2.2 Basic ”abstract” concepts The privileges attribution is not explicitly made for each user and for each object. Privileges are attributed to users according to their roles played in a given organization. Similarly, privileges are not directly related to objects and actions, but rather to classes of objects and actions. For this aim, the entities *Role*, *View* and *Activity* are introduced.

The entity *Role* is associated with subjects that fulfil same functions. The entity *View* corresponds to a set of objects that satisfy common properties. The entity *Activity* corresponds to actions that share the same principles.

2.3 Contexts *Contexts* are used to specify the circumstances where organizations grant roles permissions to perform activities on views. Contexts are defined using the relationship "$Define$" (see Figure 1). The relationship $Define(org, s, \alpha, o, c)$ means that within the organization org, context c is true between subject s, object o and action α. For instance, $Define(Purpan, John, read, F31.doc, attend\_phys)$ define the context "attending physician", within the organization $Purpan$, between subject $John$, action $read$ and object $F31.doc$.

2.4 Modelling information security A security policy is a set of permissions (resp. prohibitions) rules which are not directly defined on users, objects and actions but on their abstraction roles, views and activities. In our model, they are defined using the relationships *Permission*, *Prohibition* and *Obligation* (see Figure 2). The relationship $Permission(org, r, a, v, c)$ means that the organization org grants to a role r a permission to perform an activity a on a view v within the context c. For instance, $Permission(Purpan, phys, consulting, med\_record, attend\_phys)$ means that the Purpan hospital grants physicians permission to consult medical records within the context attending physician. The relationships $Prohibition(org, r, a, v, c)$ and $Obligation(org, r, a, v, c)$ are defined similarly.

2.5 Concrete privileges The relationships *Permission, Prohibition* and *Obligation* are not directly associated to *users, actions* and *objects* but to their respective abstractions *roles, activities* and *views*. Therefore, 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*, *Is-prohibited* and *Is-obliged* are introduced. The relationship $Is\_permitted(s, \alpha, o)$ means that subject s is permitted to perform action α on object o. *Is-prohibited* and *Is-obliged* are defined in a similar way (see Figure 3). As we will see in the next section, these relationships are logically derived from abstract permissions, prohibitions and obligations granted to roles, views and activities.

2.6 From abstract to concrete privileges To jump from abstract permissions (resp. prohibitions and obligations) to concrete permissions (resp. prohibitions and obligations), we need to relate subjects to roles, objects to views and actions to activities. This is obtained using the relationships *Employ(org,s,r)*, *Use(org,o,v)* and *Consider(org,α,a)*, as it is illustrated in Figure 4.

The relationship *Employ(org,s,r)* means that the organization org employs a subject s in a role r. For instance, $Employ(Purapn, John, phys)$ means that the Purpan hospital employs $John$ as a $physician$.

Similarly, the relationship *Use(org,o,v)* means that the organization org uses the object o in a view v, and the relationship $Consider(org, \alpha, a)$ means that the organization org considers that action α falls within the activity a.

3 First-order logical formalization of OrBAC

Logical-based approaches offer a natural way to formalize security policies. They provide tools for inference process and computing conflicting pieces of information. In this section, we present a first order logic encoding of OrBAC system. First order logic is enough since, in OrBAC system, deontic modalities (*Permission, Prohibition* and *Obligation*) only bear on elementary actions (there is no disjunctions of deontic modalities, no nested modalities,...).

3.1 Different components of OrBAC knowledge base

Available information are encoded by means of first order knowledge bases, composed of:

3.1.1 Set of facts Facts can be of different types:

- facts encoding the OrBAC relationships (except *Define*): they relate to the relationships described in the previous section (i.e., relationships of the relational diagram of Figures 1,2,3 and 4) except the *Define* relationship which is not given as a fact. For instance, we can have:
  $Employ(Purpan, Mary, nurse)$ (i.e. the Purpan hospital employs Mary as a nurse).

- facts related to the OrBAC entities attributes: these facts give information concerning the attributes of the OrBAC entities. For instance, $on\_strike(John)$ means that the subject $John$ is on strike.

3.1.2 Set of completely certain rules Completely certain rules can be of three types:

- inheritance rules: Roles, views and activities are generally organized in hierarchies. Rules of hierarchies between roles are written as: $\forall s, Employ(org, s, r_1) \rightarrow Employ(org, s, r_2)$. For instance, physicians in the Purpan hospital are also members of medical staffs, is written as: $\forall s, Employ(Purpan, s, phys) \rightarrow Employ(Purpan, s, med\_staff)$.

  Roles hierarchies imply permissions inheritance. For example, physicians will inherit all the permissions associated to role "medical staff", namely: $\forall a \forall v \forall c, Permission(org, r_2, a, v, c) \rightarrow Permission(org, r_1, a, v, c)$. Hierarchies between activities and views are defined in a similar way.

- rules related to contexts: Contrary to the other relationships of Figures 1,2,3 and 4 which are given as facts, the relationship $Define(org, s, \alpha, o, c)$ is generally described by means of first order formulas expressing conditions on which the context c is true. These formulas constraint attributes of organization org, action α, object o and subject s. For instance, the context *on-strike* is given by: $\forall s \forall \alpha \forall o$, $Define(Purpan, s, \alpha, o, strike) \leftrightarrow on\_strike(s)$: in the Purpan hospital, the context $strike$ is true between subject s, object o and action α if and only if s is on strike.

- modalities' relationships: In the rest of the paper, "obligations" will be omitted for sake of simplicity. The following axiom links concrete permissions *Is-permitted* to concrete prohibitions *Is-prohibited*: $\forall s \forall \alpha \forall o, \neg Is\_permitted(s, \alpha, o) \vee \neg Is\_prohibited(s, \alpha, o)$, which means that a user cannot be permitted and prohibited to execute the same action on the same object. Note that we do not assume equivalence between $Is\_permitted$ and $\neg Is\_prohibited$.

3.1.3 Set of uncertain rules Uncertain rules are rules which allow to jump from abstract permissions $Permission(org, r, a, v, c)$ (resp. prohibitions), between roles, activities and views, into concrete permissions $Is\_permitted(s, \alpha, o)$ (resp. $Is\_prohibited$), between subjects, actions and objects. This transformation is given by the following axiom (for each context c): $\forall org \forall s \forall \alpha \forall o \forall r \forall a \forall v$ $Permission(org, r, a, v, c) \wedge Employ(org, s, r) \wedge Use(org, o, v) \wedge Consider(org, \alpha, a) \wedge Define(org, s, \alpha, o, c) \rightarrow Is\_permitted(s, \alpha, o)$:

If the organization org, within the context c, grants role r permission to perform activity a on view v, and if org employs subject s in role r, and if org uses object o in the view v, and if org considers that action α falls within the activity a and if, within the organization org, the context c is true then s is permitted to perform α on o.
The transformation from abstract prohibition to concrete prohibition is defined in a similar way.
Note that only the context is not universally quantified. To take into account the hierarchies between roles, activities and views, these rules should also be instantiated with respect to each role, activity and view.
For sake of simplicity, we use the notation
$\phi_P(org, r, a, v, c, s, o, \alpha)$
$\equiv Permission(org, r, a, v, c) \wedge Employ(org, s, r) \wedge Use(org, o, v) \wedge Consider(org, \alpha, a) \wedge Define(org, s, \alpha, o, c) \rightarrow Is\_permitted(s, \alpha, o)$.
Similarly, we define $\phi_I(org, r, a, v, c, s, o, \alpha) \equiv Prohibition(org, r, a, v, c) \wedge Employ(org, s, r) \wedge Use(org, o, v) \wedge Consider(org, \alpha, a) \wedge Define(org, s, \alpha, o, c) \rightarrow Is\_prohibited(s, \alpha, o)$.

Example 1 *Let us suppose that we have one organization the Purpan hospital, one patient JO, a physician $John$ who is supposed to be one of JO's attending physicians, the activity $consulting$ which relates to the action $read$, the view "$medical\ record$" which contains $JO's\ medical\ record$. Suppose that the Purpan security policy contains the following two rules:*

1. *a physician has a permission to consult the medical record of his patient.*
2. *a physician, who is on strike, has a prohibition to consult patients' medical records.*

Using these two rules, we extract one role "physician", and two contexts "attending physician" and "strike". Suppose that $John$ is on strike. The encoding of these rules is as follows:

Set of facts:
$\mathbf{F_1}$. $Permission(Purpan, phys, consulting, med\_record, attend\_phys)$.
$\mathbf{F_2}$. $Prohibition(Purpan, phys, consulting, med\_record, strike)$.
$\mathbf{F_3}$. $Consider(Purpan, read, consulting)$.
$\mathbf{F_4}$. $Use(Purpan, med\_record\_JO, med\_record)$.
$\mathbf{F_5}$. $Employ(Purpan, John, phys)$.
$\mathbf{F_6}$. $Name(med\_record\_JO) \in Patient(John)$.
$\mathbf{F_7}$. $on\_strike(John)$.

Set of completely certain rules:
$\mathbf{R_1}$. $\forall s \forall \alpha \forall o$
$\neg Is\_permitted(s, \alpha, o) \vee \neg Is\_prohibited(s, \alpha, o)$.

$\mathbf{R_2}$.
$\forall s, Define(Purpan, s, read, med\_record\_JO, attend\_phys) \leftrightarrow Name(med\_record\_JO) \in Patient(s)$.
$\mathbf{R_3}$.
$\forall s, Define(Purpan, s, read, med\_record\_JO, strike) \leftrightarrow on\_strike(s)$.

Set of uncertain rules:
$\mathbf{URP_1}$. $\phi_P(org, r, a, v, attend\_phys, s, o, \alpha)$.
$\mathbf{URP_2}$. $\phi_P(org, r, a, v, strike, s, o, \alpha)$.
$\mathbf{URI_1}$. $\phi_I(org, r, a, v, attend\_phys, s, o, \alpha)$.
$\mathbf{URI_2}$. $\phi_I(org, r, a, v, strike, s, o, \alpha)$.

From the set of facts $\{F_1, F_3, F_4, F_5, F_6\}$ and the set of rules $\{R_2, URP_1\}$, we deduce that $John$ is permitted to read JO's medical record. While, by using the set of facts $\{F_2, F_3, F_4, F_5, F_6, F_7\}$ and the set of rules $\{R_3, URI_2\}$, we deduce that it is prohibited to $John$ to read JO's medical record . This leads to a conflict (using rule R_1).

4 Handling dynamic security policies

This section deals with the problems of revision of security policies. Revising security policies is crucially important for sound decision making and effective communication. Policy revision can refer to addition or suppression of regulations. As it is stated in the above sections, existing access control models are not suitable for handling evolutions of security policies and systems. Introducing new regulations poses a problem when the new rules contradict or question existing regulations. Similarly, the suppression of existing rules can be problematic that requires to inspect the intention of the modification. This section presents the possibilistic logic approach.

4.1 Possibilistic Logic

Let L be a finite propositional language. $\vdash$ denotes the classical consequence relation. Ω is the set of classical interpretations or worlds, and $[\phi]$ is the set of classical models of ϕ. Often epistemic states (or cognitive states), viewed as a set of beliefs about the real world (based on the available information), are represented by a total pre-order either on Ω, or on the set of formulas. These orderings reflect the strength of the various knowledge maintained by an agent. A priority ordering over knowledge can be encoded using different types of scale: a finite linearly ordered scale, the integers (possibly completed by $+\infty$), the unit interval [0,1], etc. In this section, we describe the representation of epistemic states in possibilistic logic both at the syntactic and semantic level, where priorities are encoded by reals in the interval [0,1].

4.2 Semantic representation

A possibility distribution π is a mapping from Ω to the interval [0,1]. $\pi(\omega)$ represents the degree of compatibility of ω with the available information (or beliefs) about the real world. $\pi(\omega) = 0$ means that the interpretation ω is impossible, and $\pi(\omega) = 1$ means that nothing prevents ω from being the real world. The interpretations such that $\pi(\omega) = 1$ are considered as normal, expected. When $\pi(\omega) > \pi(\omega')$, ω is a preferred candidate to ω' for being the real state of the world. The less $\pi(\omega)$ the more abnormal ω is. A possibility distribution π is said to be normal if $\exists\omega \in \Omega$, such that $\pi(\omega) = 1$.

Given a possibility distribution π, we can define two different measures on formulas of the language:

- the possibility degree $\Pi_\pi(\phi) = \max\{\pi(\omega) : \omega \in [\phi]\}$ which evaluates the extent to which ϕ is consistent with the available information expressed by π.
- the necessity degree $N_\pi(\phi) = 1 - \Pi(\neg\phi)$ which evaluates the extent to which ϕ is entailed by the available information.

When there is no ambiguity, we simply write $\Pi(\phi)$ (resp. $N(\phi)$) instead of $\Pi_\pi(\phi)$ (resp. $N_\pi(\phi)$). Note that $\Pi(\phi)$ is evaluated from the assumption that the situation where ϕ is true is as normal as can be. The duality equation $N(\phi) = 1 - \Pi(\neg\phi)$ extends the one existing in classical logic, where a formula is entailed from a set of classical formulas if and only if its negation is inconsistent with this set.

4.3 Syntactic representation of epistemic states

A possibility distribution can also be represented syntactically by means of possibilistic knowledge bases which are made of a finite set of weighted formulas

$$\Sigma = \{(\phi_i, a_i), i = 1, n\}$$

where a_i is understood as a lower bound of the degree of necessity $N(\phi_i)$ (namely $N(\phi_i) \geq a_i$). Formulas with null degree are not explicitly represented in the knowledge base (only beliefs which are somewhat accepted by the agent are explicitly represented). The higher the weight, the more certain the formula.

Definition 1 *Let Σ be a possibilistic knowledge base, and a∈[0,1]. We call the a-cut of Σ (resp. strict a-cut), denoted by $\Sigma_{\geq a}$ (resp. $\Sigma_{>a}$), the set of classical formulas in Σ having a certainty degree at least equal (resp. strictly greater than) a.*

A possibilistic knowledge base Σ is said to be consistent if the classical knowledge base, obtained by forgetting the weights, is classically consistent. Each inconsistent possibilistic base is associated with a level of inconsistency in the following way:

Definition 2 *Let Σ be a possibilistic knowledge base. The inconsistency degree of Σ is:*

$$Inc(\Sigma) = Max\{a : \Sigma_{\geq a} \text{is inconsistent}\}$$

with Max(∅)=0.

Given a possibilistic belief base Σ, we can generate a possibility distribution from Σ by associating to each interpretation, its level of compatibility with agent's beliefs, namely with Σ, as explained now.

When a possibilistic belief base is only made of one formula $\{(\phi, a)\}$, then each interpretation ω which satisfies ϕ gets a possibility degree $\pi(\omega) = 1$ (since it is completely consistent with ϕ) and each interpretation ω which falsifies ϕ gets a possibility degree $\pi(\omega)$ such that the higher a is (i.e., the more certain ϕ is), the lower $\pi(\omega)$ is. In particular, if $a = 1$ (i.e., ϕ is completely certain), then $\pi(\omega) = 0$, namely ω is impossible if ω falsifies ϕ. One way to represent this constraint is to assign to $\pi(\omega)$ the degree $1 - a$ with a numerical encoding. More generally if π takes its value on a linearly ordered scale, $1 - (\cdot)$ is to be understood as an order-reversing map of the scale. Therefore, the possibility distribution associated to $\Sigma = \{(\phi, a)\}$ is:

$$\begin{aligned} \forall\omega \in \Omega, \pi_{\{(\phi,a)\}}(\omega) &= 1 \quad \text{if } \omega \models \phi \\ &= 1 - a \quad \text{otherwise.} \end{aligned}$$

When $\Sigma = \{(\phi_i, a_i), i = 1, n\}$ is a general possibilistic belief base then all the interpretations satisfying all the beliefs in Σ will have the highest possibility degree, namely 1, and the other interpretations will be ranked w.r.t. the highest belief that they falsify, namely we get [Dubois et al.1994]:

Definition 3 *: The possibility distribution associated with a knowledge base Σ is defined by:*

$$\forall\omega \in \Omega, \pi_\Sigma(\omega) = \begin{cases} 1 & \text{if } \forall(\phi_i, a_i) \in \Sigma, \omega \in [\phi_i] \\ 1 - max\{a_i : (\phi_i, a_i) \in \Sigma \text{ and } \omega \notin [\phi_i]\} & \text{otherwise.} \end{cases}$$

Example 2 *Let $\Sigma = \{(q, .3), (q \vee r, .5)\}$. Then:*

ω	$\pi_\Sigma(\omega)$
qr	1
$q\neg r$	1
$\neg qr$	.7
$\neg q\neg r$	.5

The two interpretations qr and $q\neg r$ are the preferred ones since they are the only ones which are consistent with Σ, and $\neg qr$ is preferred to $\neg q\neg r$, since the highest belief falsifi ed by $\neg qr$ (i.e. $(q, .3)$) is less certain than the highest belief falsifi ed by $\neg q\neg r$ (i.e. $(q \vee r, .5)$).

4.4 Revision in possibilistic logic

Belief revision results from the effect of accepting a new piece of information called the input information [Gärdenfors1988, Nebel1994, Papini2001].

In this paper, we consider the revision with a totally reliable (or certain, sure) input p. This means that all interpretations that falsify p are declared impossible. This is performed by means of a conditioning device which transforms a possibility distribution π and a new and totally reliable information p to a new possibility distribution denoted by $\pi' = \pi(.|p)$.

The well-known type of conditioning [Dubois and Prade1998] is: (when $\Pi(p) > 0$):

$$\begin{aligned} \pi(\omega \mid_m p) &= 1 \text{ if } \pi(\omega) = \Pi(p) \text{ and } \omega \models p \\ &= \pi(\omega) \text{ if } \pi(\omega) < \Pi(p) \text{ and } \omega \models p \\ &= 0 \text{ if } \omega \notin [p]. \end{aligned}$$

This is the definition of *minimum-based conditioning.* Besides, when $\Pi(p) = 0, \pi(\omega \mid_m p) = 1, \forall\omega$, by convention.

Example 3 *Let us revise the possibility distribution π_Σ given in Example 1 by the information that q is certainly false. If we use minimum-based conditioning we get:*

ω	$\pi_\Sigma(\omega \mid_m \neg q)$
$\neg q\, r$	1
$\neg q\, \neg r$	.5
qr	0
$q\, \neg r$	0

A syntactic counterpart of revising with totally reliable information consists of constructing from a possibilistic base Σ and the new information p, a new possibilistic base Σ' such that:

$$\forall\omega, \pi_{\Sigma'}(\omega) = \pi_\Sigma(\omega|p).$$

where | is the minimum-based conditioning.

The construction of Σ' can be performed as follows:

- add the input p to the belief base with highest possible priority (namely 1);
- compute the level of inconsistency $x = \text{Inc}(\Sigma \cup \{(p, 1)\})$ of the resulting possibly inconsistent belief base (when $x = 1$ the revision simply acknowledges the input formula $\{(p, 1)\}$);
- drop all formulas with priority less than or equal to this level of inconsistency.

This guarantees that the remaining beliefs are consistent with p. Concerning the weights of remaining beliefs, it leaves them unchanged.

The following proposition gives the formal expression of Σ':

Proposition 1 *Let Σ be a possibilistic base and π_Σ be the possibility distribution associated with Σ (in the sense of Definition 3). Let p be new sure information and $x = Inc(\Sigma \cup \{(p, 1)\})$. Then: the possiblistic base associated with $\pi_\Sigma(\omega|_m p)$ is:*

$$\Sigma' = \{(\phi, b) : (\phi, b) \in \Sigma \; and \; b > x\} \cup \{(p, 1)\}.$$

Note that the computation of the resulting base is efficient. Its complexity is the same as the one of computing the inconsistency level of a possibilistic base. This can be done in $log_2 m$ satisfiability tests, where m is the number of layers in Σ. Moreover, the size of the resulting base is at most equal to $\mid \Sigma \mid +1$ which is reached when the input is consistent with the possibilistic base.

The contraction of a possibility distribution with respect to p corresponds to forgetting that p (for instancec a regulation) is believed. In such a case, the result π_p^- of the contraction must lead to a possibility measure Π_p^- such that $\Pi_p^-(p) = \Pi_p^-(\neg p) = 1$, i.e. complete ignorance about p. Intuitively if $\Pi(p) = \Pi(\neg p) = 1$ already exists, then we should have $\pi_p^- = \pi$. Besides if $p \in BS(\pi)$ then we should have $\pi_p^-(\omega) = 1$ for some $\omega \models \neg p$, and especially for those ω such that $\Pi(\neg p) = \pi(\omega)$. If $\Pi(\neg p) = 1 > \Pi(p)$, i.e. $\neg p$ represents an accepted belief, π should be unchanged. It leads to [Dubois and Prade1992]:

$$\begin{aligned} \pi_p^-(\omega) &= 1 \text{ if } \pi(\omega) = \Pi(\neg p) \text{ and } \omega \models \neg p \\ &= \pi(\omega) \text{ otherwise.} \end{aligned}$$

Hence, revising a security policy A, for instance by adding a permission P, is a new security policy which contains P and includes as much of initial security policy as consistently possible. If P is coherent with A, then the revision of A by P, is a simple addition of P to A.

5 Conclusions

This paper has proposed a generic tool for revising security policies, based on a possibilistic logic framework. Possibilistic logic offers several advantages. It allows the addition and suppression of regulations. It guarantees that the obtained security policies is always free of conflicts. Lastly, its computational complexity is not very high.

Acknowledgements: This work is supported by the cooperation program PAI CMEP Number 06-MDU687.

References:

[Aboulkalem et al2003] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Mi ege, C. Saurel, and G. Trouessin. Organisation based access control. In Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (Policy 2003), pages 120131, Lake Como, Italy, June 2003. IEEE Computer Society.

[Dubois et al.1994] Dubois D., Lang J., Prade H. Possibilistic logic. *In: Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 3, (D. Gabbay *et al.*, eds.) 439-513, 1994.

[Dubois and Prade1998] Dubois D., Prade H. Possibility theory: qualitative and quantitative aspects. In: *Handbook of Defeasible Reasoning and Uncertainty Management Systems* (D. Gabbay, Ph. Smets, eds.) Vol. 1: *Quantified Representation of Uncertainty and Imprecision* (Ph. Smets, ed.) , 169-226, Kluwer Academic Press, 1998.

[Dubois and Prade1992] Dubois D., Prade H. Belief change and possibility theory. In: *Belief Revision*, (P. Gärdenfors, ed.), 142-182, 1992.

[Gärdenfors1988] Gärdenfors P. *Knowledge in Flux - Modeling the Dynamic of Epistemic States*, MIT Press, 1988.

[Nebel1994] Nebel B. Base revision operator and schemes: semantics representation and complexity. *Proceedings of 11th European Conference on Artificial Intelligence (ECAI'94)*, 341-345, 1994.

[Papini2001] Papini O. Iterated revision operations stemming from the history of an agent's observations. In *Frontiers of Belief Revision*, H. Rott and M. Williams, eds. 2001.