A Policy Language for Modelling Recommendations*

Anas Abou El Kalam and Philippe Balbiani

Université de Toulouse, IRIT
firstname.lastname@irit.fr

Abstract. While current and emergent applications become more and more complex, most of existing security policies and models only consider a yes/no response to the access requests. Consequently, modelling, formalizing and implementing permissions, obligations and prohibitions do not cover the richness of all the possible scenarios. In fact, several applications have access rules with the recommendation access modality. In this paper we focus on the problem of formalizing security policies with recommendation needs. The aim is to provide a generic domain-independent formal system for modelling not only permissions, prohibitions and obligations, but also recommendations. In this respect, we present our logic-based language, the semantics, the truth conditions, our axiomatic as well as inference rules. We also give a representative use case with our specification of recommendation requirements. Finally, we explain how our logical framework could be used to query the security policy and to check its consistency.

1 Problem Statement

Authorization aims at allowing legitimate actions: it forbids non-authorized users to carry out actions and forbids internal users to carry out non-authorized actions. Basically, in order to define authorized actions, we should establish a security policy. The Common Criteria define an "*organizational security policy*" as: *a set of security rules, procedures, or guidelines imposed by an actual or hypothetical organization in the operational environment* [1]. Such an organizational security policy usually relies on an *access control policy* [2]. The latter is generally specified through: (1) the security objectives that must be satisfied, e.g., "*classified information must remain secret*"; and (2) the rules expressing how the system may evolve in a secure way, e.g., "*the owner of an information is allowed to grant a read access right on the information to other users*". An access control model is often used to rigorously specify and reason on the access control policy (e.g., to verify its consistency).

Unfortunately, while security models play an important role in any system, most researches on this topic are based on limited concepts, and do not capture all the richness of current and emergent applications. In particular, most of traditional policies are static and only make yes/no decisions in response to user requests.

Recently, several works was intended to model obligations [3] [4] [5] [6]. However, up to our knowledge, there is no existing work on recommendations, while this notion became extremely important in real applications. If we take health care systems as an example, most of the current regulations are in fact recommendations or guidelines: recommendations of the General Assembly of United Nations [7], Recommendations of the Council of Europe [8] [9], Guidelines of the European Parliament [10], etc.

* This work is supported by the ADCN Airbus contract and by the European NoE NewCom+.

D. Gritzalis and J. Lopez (Eds.): SEC 2009, IFIP AICT 297, pp. 176–189, 2009.

Similarly, in the critical infrastructures area, organizations such as the European Councils [11], the International Risk Governance Council (IGRC) [12], the North American Electric Reliability Council (NERC), etc. state several recommendations to protect these infrastructures (e.g., Electrical power grid) [13]. In these legislation and documents, we find rules such as: "*it is recommended that ...*", "*it is inadvisable that ...*".

However, while security policies should translate these recommendations to security rules, there is no logical framework that helps to adequately formalize this task. Basically, when building systems, we need firstly to precisely specify the underlying requirements (e.g., recommendations); and secondly, we need axioms, methods and tools for reasoning on these concepts. To date, these problems have not been really addressed. Dealing whit these issues, this paper is organized as follow. In Section 2, we discuss the security requirements already handled by classical security policies and models. After that, Section 3 defines the new recommendation access control modality. Then, Section 4 presents our new logical-based framework for modelling recommendations. In particular, we will define our new Recommendation language (RL), the related semantics, truth conditions and axiomatic. Then, Section 5 describes some ideas to query the security policy and to verify its consistency / coherence. Finally, Section 6 draws conclusions an perspectives.

2 Traditional Security Policies and Models

A security policy specifies, usually in a textual form, who has access to what, when and in which conditions? Nevertheless, the security policy does not guarantee a secure and correct functioning of the system. The security policy can indeed be badly designed or intentionally / accidentally violated. Consequently, it is important to associate a model to it; this kind of "precise statement" helps to: abstract the policy and handle its complexity; represent the secure states of a system (i.e., states that satisfies the security objectives) as well as the way in which the system may evolve (the possible executions of the system); verify the coherence of the security policy and detect possible conflicting situations (e.g., situation where a certain user has the recommendation (or the permission) and the prohibition to carry out a certain action on the same object); guarantee that all the security objectives are covered by the security mechanisms implementing the policy; etc.

We can assert that, until now, it is not possible to explicitly specify recommendations in existing access control models (e.g., discretionary "DAC" [14] [15], mandatory access control "MAC" [16] and Role-based Access Control "RBAC" [17]). For instance, the HRU model [15] represents with a matrix $M(s, o)$ the actions that a subject s is allowed to carry out on an object o. Similarly, in Role Based-Access Control (RBAC) roles are assigned to users, permissions are assigned to roles and users acquire permissions by playing roles [17].

Besides that, some works have addressed the notion of explicit prohibitions and obligations. For example, in the OrBAC model [18], security rules have the form *AccessModality* (*org*; *r*; *v*; *a*; *c*); while *AccessModality* is a Permission, Obligation or a Prohibition. This rule means: in the context *c*, organization *org* grants role *r* the permission or the obligation or the prohibition to perform activity *a* on view *v*.

In XACML [19], obligations are a set of operations that must be fulfilled in conjunction with an authorization decision (permit or deny).

Bettini *et* al. distinguish between *provisions* and *obligations* [3]. Provisions are conditions that need to be satisfied or actions that must be performed before a decision is rendered, while obligations are actions that must be fulfilled by either the users or the system after the decision.

Hilty *et* al. Define the OSL, an Obligation Specification Language that allows formulating a wide range of usage control requirements [6]. They differentiate between usage and obligational formulae. Usage is concerned with operations (e.g., processing, rendering, execution, management, or distribution) on data that must be protected; while obligational formulae are conditions on the usage of data, e.g., "delete document D within 30 days". An obligational formula becomes an obligation once a data consumer is obliged to satisfy it, i.e., once the data consumer has received the data and committed to the condition.

3 The Recommendation Access Modality

By modelling permissions, obligations and prohibitions, traditional access control policies and models control who can (permission), must (obligation) and cannot (prohibition) access to data respectively. However, these access modalities do not deal with situations where the system interact with the user by advising him (not obliging him) to do something, and if the user does not follow this advise, he/she accepts the consequences of his/her action. In this respect, it seems interesting to consider an access modality that is stronger than permissions but not as restricting as obligations. This new modality is actually a *recommendation*.

For example, the law [20] gives patients the right to access their medical files, but it recommends that this access be done through the attending physician (because certain notions in the medical file could be badly understood by the patient, while the physician can understand and explain correctly the situation). The same law stipulates that if in addition the patient is minor or suffers from psychological disorders, it is *recommended* that he/she be accompanied with his/her tutor.

In fact, we see that this access is stronger than permissions (as the patient accepts the consequences if he/she does not respect the recommendation) but not as restricting as obligations (as he/she is not obliged to respect the recommendation, i.e., he/she can access his/her medical file).

Let us take another example, the Council of Europe Recommendation No. R (97) 5 "*on the Protection of Medical Data*" [9]. This legislation recommends that medical data shall be obtained from the data subject. It is not an obligation, as medical data can be obtained from other sources in certain situations (e.g., in particular if the data subject is not in a position to provide the required data). And in the same time, this access is stronger than a permission, as the data subject could ask for explanation / justification if the recommendation is not respected, and in certain situations he/she can contest before the judge.

In the same sense, some organizations (e.g., the Computer Emergency Readiness Team "CERT", the World Wide Web Consortium "W3C") and constructors (e.g., CISCO) regularly publish recommendations [21] [22] [23]. Moreover, in the Internet field for example, the IETF associate the "*Should*" verb to a "*recommendation requirement*" in the specification of standard track documents [24]. More precisely, the RFC

2119 states that: "*must*", "*required*" or "*shall*" mean that the definition is an absolute requirement of the specification; "*must not*" or "*shall not*" mean that the definition is an absolute prohibition; "*should*" or the adjective "***recommended***" mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course; "*should not*" or "***not recommended***" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label. We can give several other examples, but due to space limitation we can conclude that security policies in many applications became more and more complex, and there is a great need to find mechanisms to handle the concept of recommendation. This is a big challenge that has never been addressed.

In this paper, we believe that the recommendation notion is halfway between permissions and obligations (i.e., recommendations are stronger than permissions but not as restricting as obligations); in the same way, inadvisabilities seem halfway between prohibited and elective (cf. next Section) actions (i.e., inadvisabilities are weaker than prohibitions but stronger than elective actions). The purpose of the two next sections is to present a logical framework that provides a means of specifying and reasoning about permissions, prohibitions, obligations, recommendations (e.g., should) and inadvisabilities (e.g., should not ...) in a given universe of entities.

4 Modelling Recommendations

Roughly speaking, the choice of a formal language for specifying a security policy is based, on one hand, on the expressive power of this language and, on the other hand, on the requirements of the targeted applications. Moreover, in order to specify the security policies that interest us in this paper, we need first to express norms, i.e. rules which say what *must* be the case, *must not* be the case, *may be* the case or *may not be* the case. Actually, this kind of notions (may, must, ...) was already addressed by several logical models such as deontic logic. The latter can be seen as an extension of modal logic that considers modal operators such as obligations, permissions and prohibitions. Note that researches in deontic reasoning within a modal logic point of view has already been done by several works such as by Aqvist [25] and Prior [26]. Moreover, within the context of computer security, several authors like Bieber and Cuppens [27], Glasgow et al. [28], Prakken and Sergot [29], etc. have used deontic logic.

In the rest of the following sub-sections, we progressively extend the modal logic in order to model the notions of "recommendation" and "inadvisabilities".

4.1 Syntax

Let PV be a countable set of propositional variables, with typical members denoted p, q, etc. By means of the Boolean operators $\neg$ ("not ...") and $\vee$ ("... or ...") of classical logic and the modal operator $\mathbf{O}$ ("it is obligatory that ...") of modal logic, we combine these variables so as to build up the set of formulas of deontic logic given by the rule:

- $\phi ::= p \mid \neg\phi \mid (\phi \vee \phi) \mid \mathbf{O}\phi$.

We make use of the standard abbreviations for the other Boolean operators. We supplement the language by the modal operators **F**, **P**, and **E** expressing "*it is forbidden that* …", "*it is permitted that* …", and "*it is elective that* …": $\mathbf{F}\phi = \mathbf{O}\neg\phi$, $\mathbf{P}\phi = \neg\mathbf{O}\neg\phi$, $\mathbf{E}\phi = \neg\mathbf{O}\phi$. Basically, the specific characteristic of a norm is the consistency of the set of all obligations that make it up. This characteristic corresponds to the formula $\neg(\mathbf{O}\phi \wedge \mathbf{O}\neg\phi)$. Seeing that the "obligatory that" is the "forbidden that not" and the "forbidden that" is the "obligatory that not"; this characteristic also corresponds to the formulas $\neg(\mathbf{F}\phi \wedge \mathbf{F}\neg\phi)$ and $\neg(\mathbf{O}\phi \wedge \mathbf{F}\phi)$.

Furthermore, using the equivalences $\neg\mathbf{O}\neg\phi \leftrightarrow \mathbf{E}\neg\phi$ and $\neg\mathbf{F}\neg\phi \leftrightarrow \mathbf{P}\neg\phi$, we can deduce that $\mathbf{O}\phi \rightarrow \mathbf{E}\neg\phi$ and $\mathbf{F}\phi \rightarrow \mathbf{P}\neg\phi$. The modal operators **P** ("*it is permitted that* …") and **E** ("*it is elective that* …") keep up similar relations: the "permission that" is the "elective that not" and the "elective that" is the "permission that not". Hence, we can deduce the following formulas $\mathbf{O}\phi \rightarrow \mathbf{P}\phi$ and $\mathbf{F}\phi \rightarrow \mathbf{E}\phi$.

However, none of the previous modalities is able to directly capture the notion of "*recommendation*". Subsequently, we introduce the modal operator **R** ("*it is recommended that* …") and we use it to extend the previous set of deontic logic formulas. In fact, let us now consider the set of formulas given by the rule:

- $\phi ::= p \mid \neg\phi \mid (\phi \vee \phi) \mid \mathbf{O}\phi \mid \mathbf{R}\phi$.

Let us take a simple example. If we assume that (Read, Bob, UserGuide) is a formula expressing the fact that Bob read the user guide, in our language we can express formulas such as **R**(Read, Bob, UserGuide); meaning that: it is recommended that Bob read the user guide.

Moreover, to be able to express rules / sentences such as "*it is inadvisable that* …", we supplement the language by the modal operator **I**: $\mathbf{I}\phi = \mathbf{R}\neg\phi$. E.g., the formula **I**(Execute, Bob, OldVersion) means that executing the old version of the program is inadvisable; i.e., it is recommended to not execute the old version.

In this respect, our new set of formulas allows us to give an account of the *consistency* of a set of recommendations by means of the formula $\neg(\mathbf{R}\phi \wedge \mathbf{R}\neg\phi)$. In fact, seeing that the "recommended that" is the "inadvisable that not" and the "inadvisable that" is the "recommended that not", this formula corresponds to the following formulas $\neg(\mathbf{I}\phi \wedge \mathbf{I}\neg\phi)$ and $\neg(\mathbf{R}\phi \wedge \mathbf{I}\phi)$: it is not possible that something being both recommended and inadvisable. The question that arises now is: what are the relations between the "obligatory that", the "recommended that" and the "permitted that" on one hand, and the "forbidden that", the "inadvisable that" and the "elective that" , on the other hand. The semantics and the axiomatics of the two next subsections will allow us to show, among others, that the formulas $\mathbf{O}\phi \rightarrow \mathbf{R}\phi$, $\mathbf{R}\phi \rightarrow \mathbf{P}\phi$, $\mathbf{F}\phi \rightarrow \mathbf{I}\phi$ and $\mathbf{I}\phi \rightarrow \mathbf{E}\phi$ express indisputable obvious deontic facts.

4.2 Semantics

The most elementary model of obligations is composed of a non-empty set W of states and a relation $\mathfrak{R}$ on W. Therefore, a *deontic frame* will be an ordered pair:

- $\mathscr{F} = (W, \mathfrak{R})$

where W is a nonempty set of states and $\Re$ is a binary relation on W called accessibility relation: for all states x, the states y such that $x\Re y$ are those states in which *all* the obligations in x are satisfied. For this reason, we may also consider that for all states x, the set $\Re(x) = \{y: x\Re y\}$ characterizes the set of all permissions in x.

Actually, the formulas of deontic logic are valued at states. The valuation of the formula $\mathbf{O}\phi$ at state x depends on the valuation of ϕ at states y such that $x\Re y$.

In this respect, a *deontic model* is an ordered triple:

- $\mathscr{M} = (W, \Re, V)$

where $\mathscr{F} = (W, \Re)$ is a deontic frame and V is a valuation on W, i.e. a function assigning to each state x in W a subset $V(x)$ of the set PV of all propositional variables. $V(x)$ can thus be considered as the set of propositional variables that x verifies.

Subsequently, in the deontic model $\mathscr{M}$, the function V can be extended to the function $\bar{V}$ defined as follows:

- $p \in \bar{V}(x)$ iff $p \in V(x)$; and $\neg\phi \in \bar{V}(x)$ iff $\phi \notin \bar{V}(x)$;
- $\phi \vee \psi \in \bar{V}(x)$ iff $\phi \in \bar{V}(x)$ or $\psi \in \bar{V}(x)$;
- $\mathbf{O}\phi \in \bar{V}(x)$ iff for all states y such that $x\Re y$, $\phi \in \bar{V}(y)$.

Furthermore, according to the relationships between obligations, permissions, prohibitions (*cf.* Section 4.1), it is a simple matter to check that:

- $\mathbf{F}\phi \in \bar{V}(x)$ iff for all states y such that $x\Re y$, $\phi \notin \bar{V}(y)$,
- $\mathbf{P}\phi \in \bar{V}(x)$ iff for some state y with $x\Re y$, $\phi \in \bar{V}(y)$,
- $\mathbf{E}\phi \in \bar{V}(x)$ iff for some state y with $x\Re y$, $\phi \notin \bar{V}(y)$.

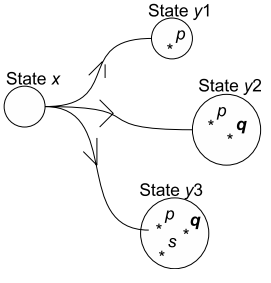

Fig. 1. A model with recommendations

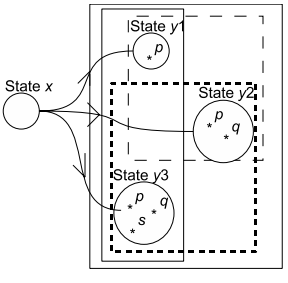

Fig. 2. Exemple of a large subset

In the model given in Fig. 1, p is obligatory at state x, whereas q and s are only permitted.

Let us now define the notions of "satisfiability" and "validity" in our model. Let ϕ be any formula. We say that ϕ is *valid* in the model $\mathscr{M} = (W, \Re, V)$ iff $\phi \in \bar{V}(x)$ for all states x; whereas ϕ is said to be valid in the frame $\mathscr{F} = (W, \Re)$ iff ϕ is valid in every model $\mathscr{M} = (W, \Re, V)$ based on $\mathscr{F}$.

Furthermore, we say that ϕ is *satisfiable* in $\mathscr{M} = (W, \Re, V)$ iff $\neg\phi$ is *not valid* in $\mathscr{M} = (W, \Re, V)$; whereas ϕ is said to be satisfiable in frame $\mathscr{F} = (W, \Re)$ iff ϕ is satisfiable in some model $\mathscr{M} = (W, \Re, V)$ based on $\mathscr{F}$.

Actually, the definitions of satisfiability and validity come from the semantics for modal logic. Correspondence theory in modal logic teaches us the ways the validity of the modal formulas $\neg(\mathbf{O}\phi \wedge \mathbf{O}\neg\phi)$, $\neg(\mathbf{F}\phi \wedge \mathbf{F}\neg\phi)$ and $\neg(\mathbf{O}\phi \wedge \mathbf{F}\phi)$ considered above is related to the condition of seriality saying that for all states x, there exists a state y such that $x\mathfrak{R}y$. For this reason, in the sequel, we will always consider that frames are fitted out with a serial relation.

Let us now focus on the modal operators $\mathbf{O}$ and $\mathbf{P}$. The reader may easily verify that in all models $\mathscr{M} = (W, \mathfrak{R}, V)$:

- $\mathbf{O}\phi \in \bar{V}(x)$ iff $\mathfrak{R}(x) \cap \{y\colon \phi \in \bar{V}(y)\} = \mathfrak{R}(x)$, i.e. $\{y\colon \phi \in \bar{V}(y)\}$ *entirely covers* $\mathfrak{R}(x)$,
- $\mathbf{P}\phi \in \bar{V}(x)$ iff $\mathfrak{R}(x) \cap \{y\colon \phi \in \bar{V}(y)\} \neq \emptyset$, i.e. $\{y\colon \phi \in \bar{V}(y)\}$ *partially covers* $\mathfrak{R}(x)$.

Seeing that we would like the formulas $\mathbf{O}\phi \rightarrow \mathbf{R}\phi$ and $\mathbf{R}\phi \rightarrow \mathbf{P}\phi$ to be valid, the interpretation of the recommendation modal operator $\mathbf{R}$ in a model $\mathscr{M} = (W, \mathfrak{R}, V)$ should actually be halfway between the interpretations of $\mathbf{O}$ and $\mathbf{P}$ (*cf.* Section 3), i.e. it should correspond to the following interpretation:

- $\mathbf{R}\phi \in \bar{V}(x)$ iff $\{y\colon \phi \in \bar{V}(y)\}$ *covers a large part of* $\mathfrak{R}(x)$.

In this respect, the interpretation of $\mathbf{I}$ in $\mathscr{M} = (W, \mathfrak{R}, V)$ should correspond to $\mathbf{I}\phi \in \bar{V}(x)$ iff $\{y\colon \phi \in \bar{V}(y)\}$ *covers a small part of* $\mathfrak{R}(x)$.

Note that the notions "*entirely cover* (obligations), *partially cover* (permissions) and *cover a large part* (recommendations) perfectly reflect that recommendations are stronger than permissions but not as restricting as obligations (cf. Section 3).

Following our reasoning, we consider that a frame for recommendation is an ordered triple:

- $\mathscr{F} = (W, \mathfrak{R}, \mathscr{N})$

where $(W, \mathfrak{R})$ is a deontic frame and $\mathscr{N}$ is a neighborhood function on W, i.e. a function assigning to each state x in W a set $\mathscr{N}(x)$ of subsets of $\mathfrak{R}(x)$. For all states x, we will think of $\mathscr{N}(x)$ as the set of large subsets of $\mathfrak{R}(x)$. Such large subsets will characterize the set of all recommendations in x.

Now, with the recommendation notion, our model is an ordered 4-tuple:

- $\mathscr{M} = (W, \mathfrak{R}, \mathscr{N}, V)$

where $\mathscr{F} = (W, \mathfrak{R}, \mathscr{N})$ is a frame for recommendation and V is a valuation on W. In this respect, the function V can be extended (in $\mathscr{M}$) to the function $\bar{V}$ as follows:

- $\mathbf{R}\phi \in \bar{V}(x)$ iff $\mathfrak{R}(x) \cap \{y\colon \phi \in \bar{V}(y)\} \in \mathscr{N}(x)$.

For example, let us consider the model $\mathscr{M} = (W, \mathfrak{R}, \mathscr{N}, V)$ given in Fig. 2 and obtained from Fig. 1 by defining $\mathscr{N}(x)$=$\{\{y1, y2\}, \{y2, y3\}, \{y1, y3\}, \{y1, y2, y3\}\}$ (Fig. 2). As the subset $\{\{y2, y3\}$ is considered as a large surset of $\mathfrak{R}(x)$, $\{\{y2, y2\} \in \mathscr{N}(x)$. Hence, q is recommended at state x. Note that q is not obligatory at x and that s is not recommended at x.

The reader may easily verify that the validity of the modal formulas $\mathbf{O}\phi \rightarrow \mathbf{R}\phi$, $\mathbf{R}\phi \rightarrow \mathbf{P}\phi$ considered above is related to the condition saying that for all states x, $\mathfrak{R}(x) \in \mathscr{N}(x)$ and $\emptyset \notin \mathscr{N}(x)$.

Seeing that we would like the formulas $\mathbf{O}\phi \rightarrow \mathbf{R}\phi$ and $\mathbf{R}\phi \rightarrow \mathbf{P}\phi$ to be *valid*; in the sequel, we always consider that frames of recommendation are fitted out with a neighborhood function $\mathscr{N}$ such that for all states x, $\mathfrak{R}(x) \in \mathscr{N}(x)$ and $\emptyset \notin \mathscr{N}(x)$. Note that in such frames, since $\mathbf{F}\phi = \mathbf{O}\neg\phi$, $\mathbf{E}\phi = \mathbf{P}\neg\phi$ and $\mathbf{I}\phi = \mathbf{R}\neg\phi$, then the formulas $\mathbf{F}\phi \rightarrow \mathbf{I}\phi$ and $\mathbf{I}\phi \rightarrow \mathbf{E}\phi$ are also valid.

4.3 Axiomatization/Completeness

The previous section presents the semantics of our specification and representation language for obligations and recommendations. This is certainly a first step in building a global and robust logical framework; but it remains not sufficient as we need a mean to derive new informations and to reason (e.g. by verification) on our language. Moreover, it seems necessary to give axioms and rules that define the relationships between the different access modalities (obligations, recommendations and permissions). To achieve these tasks and, thus, to complete our logical framework, we define in this section the axiomatic system *LR* of our Logic of Recommendation. In addition to the classical axioms of propositional logic, we define the following axioms of *LR*:

- $\mathbf{O}(\phi \rightarrow \psi) \rightarrow (\mathbf{O}\phi \rightarrow \mathbf{O}\psi)$,
- $\mathbf{O}\phi \rightarrow \mathbf{P}\phi$,
- $\mathbf{O}(\phi \leftrightarrow \psi) \rightarrow (\mathbf{R}\phi \leftrightarrow \mathbf{R}\psi)$,
- $\mathbf{O}\phi \rightarrow \mathbf{R}\phi$,
- $\mathbf{R}\phi \rightarrow \mathbf{P}\phi$.

The axiom $\mathbf{O}(\phi \rightarrow \psi) \rightarrow (\mathbf{O}\phi \rightarrow \mathbf{O}\psi)$ is called axiom (K). It corresponds to the fact that the modal operator $\mathbf{O}$ is interpreted in models by means of a binary relation.

The axiom $\mathbf{O}\phi \rightarrow \mathbf{P}\phi$ (axiom D) corresponds to the fact that in every frame $\mathscr{F} = (W, \mathfrak{R}, \mathscr{N})$, $\mathfrak{R}$ is such that for all states x, there exists a state y such that $x\mathfrak{R}y$.

Furthermore, the axiom $\mathbf{O}(\phi \leftrightarrow \psi) \rightarrow (\mathbf{R}\phi \leftrightarrow \mathbf{R}\psi)$ is new and has never been considered before within the context of deontic logic. It corresponds to the fact that the modal operator $\mathbf{R}$ is interpreted in models by means of a neighborhood function. This axiom can be easily analysed as follows: if its antecedent $\mathbf{O}(\phi \leftrightarrow \psi)$ -which says that ϕ and ψ are true in the same accessible words- is true, then the set of accessible ϕ-worlds and the set of accessible ψ-worlds are equal. In this case, its conclusion $\mathbf{O}\phi \rightarrow \mathbf{O}\psi$ must be true. Moreover, as for $\mathbf{O}\phi \rightarrow \mathbf{R}\phi$ and $\mathbf{R}\phi \rightarrow \mathbf{P}\phi$, we have seen that these axioms are related to the fact that in every frame $\mathscr{F} = (W, \mathfrak{R}, \mathscr{N})$, $\mathscr{N}$ is such that for all states x, $\mathfrak{R}(x) \in \mathscr{N}(x)$ and $\emptyset \notin \mathscr{N}(x)$.

Besides that, in addition to the classical inference rules of propositional logic, the inference rules of *LR* are: "from ϕ, infer $\mathbf{O}\phi$". It can be proved that all the formulas of the following forms are derivable from the axioms and inference rules of *LR*:

- $\mathbf{O}\phi \wedge \mathbf{O}\psi \rightarrow \mathbf{O}(\phi \wedge \psi)$,
- $\mathbf{O}\phi \wedge \mathbf{R}\psi \rightarrow \mathbf{R}(\phi \wedge \psi)$,
- $\mathbf{O}\phi \wedge \mathbf{P}\psi \rightarrow \mathbf{P}(\phi \wedge \psi)$.

These formulas obviously correspond to our intuitive notions of obligations, recommendations and permissions. The truth of the matter is that:

Proposition 1. *All formulas derivable from the axioms and inference rules of LR are valid in all frames.*

Proof. The proof can be done by induction on the length of the derivation of ϕ in *LR* that if ϕ is derivable in *LR* then ϕ is valid in all frames.

Proposition 2. *All formulas valid in all frames are derivable from the axioms and inference rules of LR.*

Proof. The proof is done by means of a canonical model construction. Let $\mathscr{M} = (W, \mathscr{R}, \mathscr{N}, V)$ be the model defined as follows:

- W is the set of all maximal *LR*-consistent sets of formulas,
- $\mathscr{R}$ is the binary relation on W such that for all x, y in W, $x\mathscr{R}y$ iff $\{\phi\colon O\phi \in x\} \subseteq y$,
- $\mathscr{N}$ is the neighborhood function such that for all x in W and for all subsets S of $\mathscr{R}(x)$, S is in $\mathscr{N}(x)$ iff there exists a formula ϕ such that $R\phi \in x$ and $S = \{y \in W\colon x\mathscr{R}y \text{ and } \phi \in y\}$,
- V is the valuation function such that for all x in W, $V(x) = \{p\colon p \in x\}$.

It can be proved that $\mathscr{R}$ is serial. Moreover, for all states x in W, $\mathscr{R}(x) \in \mathscr{N}(x)$ and $\emptyset \notin \mathscr{N}(x)$. Using a proof by induction on the complexity of the formula ϕ, one can show that for all states $x \in W$, $\phi \in x$ iff $\phi \in \bar{V}(x)$. As a result, if ϕ is a formula not derivable in *LR*, then $\neg\phi$ is *LR*-consistent and there is $x \in W$ such that $\neg\phi \in x$. Therefore, $\phi \notin x$ and $\phi \notin \bar{V}(x)$. It follows that ϕ is not valid in all frames.

Conversly, from the axioms and inference rules of *LR*, it is not possible to derive all the formulas of the following form:

- $\mathbf{R}\phi \wedge \mathbf{R}\psi \rightarrow \mathbf{R}(\phi \wedge \psi)$,
- $\mathbf{R}\phi \wedge \mathbf{P}\psi \rightarrow \mathbf{P}(\phi \wedge \psi)$,
- $\mathbf{P}\phi \wedge \mathbf{P}\psi \rightarrow \mathbf{P}(\phi \wedge \psi)$.

The cases of the second formula and the third formula can be simply explained by looking a the model given in Fig. 2 where, at state x, q is permitted/recommended, $\neg q$ is permitted and $q \wedge \neg q$ is not permitted. The case of the first formula is different. Although it is not derivable in *LR*, our intuition of the notion of recommendation could lead us to consider it as an additional axiom. Let LR^+ be the axiomatic system obtained from *LR* by adding the following formulas as axioms:

- $\mathbf{R}\phi \wedge \mathbf{R}\psi \rightarrow \mathbf{R}(\phi \wedge \psi)$.

We will say that a frame $\mathscr{F} = (W, \mathscr{R}, \mathscr{N})$ is $\cap$-stable iff for all states x in W, the set $\mathscr{N}(x)$ of all large subsets of $\mathscr{R}(x)$ is closed for the set-theoretical operation of intersection. Remark that the frame given in Fig. 2 is not $\cap$-closed.

It can be proved that:

Proposition 3. *All formulas derivable from the axioms and inference rules of LR^+ are valid in all $\cap$-stable frames.*

Reciprocally, by means of the canonical model construction mentioned above, on can show that:

Proposition 4. *All formulas valid in all $\cap$-stable frames are derivable from the axioms and inference rules of LR^+.*

Let us go further in our extension of our recommendation language. We can prove that from the axioms and the inference rules of LR^+, it is not possible to derive all the formulas of the following form:

- $\mathbf{O}(\phi \rightarrow \psi) \rightarrow (\mathbf{R}\phi \rightarrow \mathbf{R}\psi)$,
- $\mathbf{R}(\phi \wedge \psi) \rightarrow \mathbf{R}\phi \wedge \mathbf{R}\psi$.

Nevertheless, our intuition of recommendations lead us to accept such formulas:

- if ϕ implies ψ in all accessible, hence perfect, states, then one cannot recommend ϕ without recommending ψ,
- if ϕ and ψ are together recommended then they are separately recommended too.

This remark leads us to think that one should add to the axiomatic system LR^+, all formulas of the form $\mathbf{O}(\phi \rightarrow \psi) \rightarrow (\mathbf{R}\phi \rightarrow \mathbf{R}\psi)$ and all formulas of the form $\mathbf{R}(\phi \wedge \psi) \rightarrow \mathbf{R}\phi \wedge \mathbf{R}\psi$ considered above, thus obtaining the axiomatic system LR^{++}. We will say that a $\cap$-stable frame $\mathscr{F} = (W, \mathscr{R}, \mathscr{N})$ is filtered iff for all states x in W, the set $\mathscr{N}(x)$ of all large subsets of $\mathscr{R}(x)$ is closed upward, i.e.: for all subsets S, T of $\mathscr{R}(x)$, if S is in $\mathscr{N}(x)$ and $S \subseteq T$ then T is in $\mathscr{N}(x)$ too. It can be proved that

Proposition 5. *All formulas derivable from the axioms and inference rules of LR^{++} are valid in all filtered frames.*

Reciprocally, by means of the canonical model construction mentioned above, on can show that

Proposition 6. *All formulas valid in all filtered frames are derivable from the axioms and inference rules of LR^{++}.*

5 Using Our Formalism

5.1 Specification of the Security Policy

The axiomatic system defined in the last section, coupled with classical logic axioms could be used for several aims. In this section, two of the possible uses are explained: (1) query a given policy in order to know which rules apply to a given situation; and (2) Check the security policy consistency.

To achieve these tasks, it is first necessary to specify the operational rules, the security policy, and the security objectives. Operational rules are described by means of the propositional logic operators (non modal). For example, to specify that users play roles in their organizations, we can introduce the *play* predicate between the constant symbols: organizations, users and roles. An instance of this predicate could be for instance *Play*(*ToulouseUniversity*, *Bob*, *President*).

Besides that, we suggest expressing security objectives by using modal operators. For example, the $\mathbf{R}$(*Customer*, *Read*, *notice*) security objective means that it is recommended that customers read the notice. Finally, we propose expressing security rules

using modal formula with at least a non-modal clause (e.g., $f \rightarrow \mathbf{R}q$). It describes the link between the permissions, prohibitions, obligations, or recommendations and the state of the system. For example, the security rule: "'if the patient is minor, it is inadvisable that he/she read its medical file"' can be specified by: $Age(p) \prec 18 \rightarrow \mathbf{I}(p, read, MedicalFile(p))$. In this rule we have considered that p is a variable of type "'*patient*"'; *Age* (resp. *MedicalFile*) is a function that returns the age (resp. the medical file) of a certain patient).

5.2 Querying the Security Policy

Once we have specified the operational rules, the security policy, and the security objectives of the studied application, we can use our axiomatic to develop a tool which enables a user to query the security policy. For instance, let us assume that security administrator wants to know who is recommended to read a notice? This query is translated in the following logical formula: "$\exists n, Notice(n) \wedge \mathbf{R}(x, Read, n)$".

Note that there are two ways to program this formula in logical-based languages such as PROLOG. The first one lists the persons who are actually recommended to read a notice; while the second method answers by a formula which corresponds to a sufficient condition that satisfies the query. This second technique of query answering is called intentional answer in [30].

5.3 Checking the Security Policy Consistency

Different techniques can be used to check the security policy consistency, in particular, we can use:

- Axiom-based methods, called Frege-Hilbert methods. The idea is to derive new rules by applying the inference rules to the set of axioms until demonstrating the intended property. Note that it is difficult to mechanize this method since it is difficult to find the wanted property among all the possible deductions.
- Natural deduction methods: these techniques are closed to the reasoning used by mathematicians to demonstrate their theorems. In this kind of calculus, every derivation starts by some hypothesis and assumptions [31].

In our context, it is important to choose the method that (1) gives enough information about the reasons of success or failure while demonstrating a certain security property, (2) identifies the system state that is responsible (3) identifies some resident vulnerabilities in the system or a certain weakness in the security policy specification. This will greatly enhance the system security and rigorously help to refine the security objectives. For these reasons, we suggest using a constructive verification technique such as the "Tableau method" or its variant "Gentzen sequence calculus". In order to prove a certain formula ϕ, the main idea is to assume that $\neg\phi$ is true and to derive a contradiction by successively splitting up $\neg\phi$ in each of its derived sub-formulas, until obtaining a state satisfying a formula and its negation. Actually, in this method, we draw a graph where the initial node contains an initial secure state (e.g., a state where certain security objectives are true/satisfied). Then, we progressively apply some derivation rules (specific to this method). At each state we also apply one of the security rules (rules that

specify how the system can, must or should evolve). The demonstration is ended when attending a non-secure state (a state where a contradiction is detected).

The "Tableau method" can also be used to detect conflicting situations, e.g., if, from a secure state, and by applying the security rules as well as the derivation rules, we reach a state where a certain user has the permission/obligation/recommendation and the prohibition to carry out a certain action on the same object); This problem comes to draw our graph and to look for nodes where one of the following formulas are true: $\mathbf{R}p \wedge \mathbf{F}q$ or $\mathbf{P}p \wedge \mathbf{F}q$ or $\mathbf{O}p \wedge \mathbf{F}q$ or $\mathbf{I}p \wedge \mathbf{F}q$ or $\mathbf{I}p \wedge \mathbf{R}q$ or $\mathbf{I}p \wedge \mathbf{O}q$.

6 Conclusion

Thanks to its ability to specify the concepts of obligation, permission and prohibition, Deontic logic is an attractive candidate for expressing security policies. Actually, this logic was first associated to epistemic logic and used by Glasgow and McEwen to specify confidentiality policies [28]. Bieber and Cuppens used it to model the causality, non-interference and non-deducibility security property [27]. Furthermore, Deontic logic was used in any kind of systems and applications such as databases [32]. We can thus assert that Deontic logic is well adapted to capture several security properties and modalities. However, none of the existing works have studied the recommendation and inadvisable access modalities, while these concepts are unavoidable in many current and emergent applications. Several regulations are in fact in the form of recommendations and directives, and these regulations should be reflected in security policies (in the specification as well as in the implementation phases). Modeling recommendations is thus a new challenge in the security policies and models field. In this paper, we have proposed a logical framework that covers the richness of these legislations and applications. In particular, we have enhanced Deontic logic by a new Recommendation Specification Language. Moreover, in order to be able to reason on the security policy and to derive new rules, we have suggested a new recommendation-based axiomatic. The latter can be combined by classical logic axioms to provide more general reasoning mechanisms. Now, we are integrating our language in a global access control model: OrBAC (Organization-Based Access Control) [18]. In fact, the latter is well adapted to several kinds of heterogeneous, multi-organizational and distributed systems, but it suffers from its incapacity to model and reason on recommendations. With the work presented in this paper, this weakness will be overcome. We also expect applying our work to a representative case study. Finally, we will also develop mechanisms to integrate the recommendation access modality in existing tools and languages such as Prolog.

References

1. Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 1, Part 1: Introduction and general model, CCMB-2006-09-001, 86 p. (September 2006)
2. Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 1, Part 2: Security functional components, CCMB-2006-09-002, 314 p. (September 2006)

3. Bettini, C., Jajodia, S., Wang et, X.S., Wijesekera, D.: Obligation Monitoring in Policy Management. In: International Workshop, Policies for Distributed Systems and Networks (Policy), Monterey, California, pp. 2–12. IEEE Computer Society Press, Los Alamitos (2002)
4. Demeanor, N., Delay, N., Lupus, E., Sloan, M.: The Ponder Policy Specification Language. In: International Workshop Policy, Bristol, UK, pp. 18–38. IEEE Computer Society Press, Los Alamitos (2001)
5. Ni, Q., Bertino, E., Lobo, J.: An Obligation model bridging access control policies and privacy policies. In: 13th ACM SACMAT, Estes Park, CO, USA, June 11-13 (2008)
6. Hilty, M., Pretschner, A., Basin, D., Schaefer, C., Walter, T.: A policy language for distributed usage control. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 531–546. Springer, Heidelberg (2007)
7. Resolution A/RES/45/ General assembly of United Nations, Guidelines for the regulation of computerized personal data files (December 1990)
8. Recommendation of the Communication of Health Information in Hospitals, European Health Committee CDSP (92)8, Council of Europe, Strasbourg (June 1992)
9. Recommendations of the Council of Europe, R(97)5, On The Protection of Medical Data Banks, Council of Europe, Strasbourg (February 13, 1997)
10. Directive 95/46/EC of the European Parliament and of the Council of 24, On the protection of individuals with regard to the processing of personal data (October 1995)
11. European Council, Bangemann report recommendations to the EC (May 26, 1994)
12. International Risk Governance Council, Critical infrastructures at risk: Securing the European electric power system (2007)
13. North American Electric Reliability Council, Urgent action standard 1200 (2003)
14. Lampson, B.: Protection. In: 5th Princeton Symp. on Information Sciences and Systems (1971)
15. Harrison, M.A., Ruzzo, W.L., Ullman, J.D.: Protection in Operating Systems. Communication of the ACM 19(8), 461–471 (1976)
16. Bell, D.E., LaPadula, L.J.: Secure Computer Systems: Unified Exposition and Multics Interpretation, technical report, MTR 2997 Rev. 1, MITRE corp., Bedford, USA (1976)
17. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: A Proposed Standard for RBAC. ACM Tras. on Info. and System Security 4(3) (August 2001)
18. Abou El Kalam, A., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., El-Baida, R., Miège, A., Saurel, C., Trouessin, G.: OrBAC: Organization-Based Access Control. In: 4th International Workshop Policy, Come, Italy, pp. 120–131. IEEE Computer Society Press, Los Alamitos (2003)
19. OASIS, eXtensible Access Control Markup Language TC v2.0, Normative XACML 2.0 documents, `http://www.oasis-open.org/specs/index.php`
20. Law 2002-303 related to the patient's rights and to the quality of healthcare systems, Article L. 1111-7 (March 2002)
21. W3C, W3C Recommendations, `http://www.w3.org/TR`
22. CISCO, Access Control Lists: Overview and Guidelines, `http://www.cisco.com/en/US/docs/ios/11_3/security/configuration/guide/scacls.pdf`
23. Computer Emergency Response Team, CERT alerts, `http://www.cert.org`
24. Bradner, S.: RFC2119: Key words for use in RFCs to Indicate Requirement Levels, IETF (March 1997)
25. Aqvist, L.: Next and Ought, alternative foundations for Von Wright's tense-logic, with an application to deontic logic. Logique & Analyse 9, 231–251 (1966)
26. Prior, A.: The paradoxes of derived obligation. Mind 63, 64–65 (1954)
27. Bieber, P., Cuppens, F.: A definition of secure dependencies using the logic of security. In: Computer Security Foundations Workshop IV. IEEE, Los Alamitos (1991)

28. Glasgow, J., MacEwan, G., Panagaden, P.: A logic for reasoning about security. ACM Transactions on Computer Science 10, 226–264 (1992)
29. Prakken, H., Sergot, M.: Dyadic deontic logic and contrary-to-duty obligations. In: Nute, D.N. (ed.) Defeasible Deontic Logic, Synthese Library, pp. 223–262. Kluwer, Dordrecht (1997)
30. Cholvy, L., Demolombe, R.: Querying a rule base. In: First International Conference on Expert Database Systems, Charleston (1986)
31. Fitting, M.: Basic Modal Logic. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming Logic Foundations, vol. 1(5), pp. 365–448. Oxford Science Publications (1993) ISBN 0-19-853745-X
32. Cuppens, F., Demolombe, R.: A Deontic Logic for Reasoning about Confidentiality. In: Brown, M., Camo, J. (eds.) Deontic Logic, Agency and Normative Sytems