

# Reasoning with an Incomplete Information Exchange Policy

Laurence Cholvy<sup>1</sup> and Stéphanie Roussel<sup>1,2</sup>

<sup>1</sup> ONERA Centre de Toulouse  
2 avenue Edouard Belin  
31055 Toulouse, France

<sup>2</sup> SUPAERO  
10 avenue Edouard Belin,  
31055 Toulouse, France

**Abstract.** In this paper, we deal with information exchange policies that may exist in multi-agent systems in order to regulate exchanges of information between agents. More precisely, we discuss two properties of information exchange policies, that is the consistency and the completeness. After having defined what consistency and completeness mean for such policies, we propose two methods to deal with incomplete policies.

**Keywords:** completeness, information exchange policy, multi-agent system.

## 1 Introduction

Multi-agent systems provide an interesting framework for modelling systems in which some entities (atomic entities or complex ones) cooperate in order to fulfill a common task or to achieve a common goal. In order to cooperate efficiently, the entities, now called agents, have to exchange information, in particular in order to have a common view of the environment and a common understanding of the current situation.

In many systems, exchanges of information are not constrained and agents may exchange any information they want to anybody. At the opposite, in many other systems, information exchanges are ruled by a policy, in particular in order to satisfy some security constraints, like confidentiality, or efficiency constraints (broadcasting or peer-to-peer communication of relevant information). The so-called “Systems of Systems” in defense area or in civil security area [1] are instances of such multi-agent systems as well as any organisation of people and means like companies. These systems have in common that they are made of systems (human or not, atomic or not) which are geographically distributed, independently managed and which have to share information in a risky environment so that information exchanges between these systems must be compliant with a policy.

This present work deals with this last kind of systems. The illustrative example we will take all along the paper is the example of a hierarchical company with a boss and employees who exchange information relative to the materials used

in the company. These exchanges must agree with a policy which, for instance, imposes the diffusion of pertinent and useful information as soon as possible, while respecting confidentiality restrictions.

An *information exchange policy* can then be seen as a regulation the agents must satisfy and which specifies the information exchanges which are obligatory, forbidden or permitted and under which conditions. But, in order to be useful, such a policy, as any other regulation, must satisfy several properties and in particular, it must be *consistent* and *complete*.

According to [2] which studies confidentiality policies, consistency allows to avoid cases when the user has both the permission and the prohibition to know something. More generally, according to [3] and [4], which study consistency of general kind of regulations, consistency of regulation does not come to classical consistency of a set of formulas. According to this work, a regulation is consistent if there exists no possible situation in which it leads an agent to *normative contradictions* or *dilemmas* also called in [15] *contradictory conflicts* (a given behaviour is prescribed and not prescribed, or prohibited and not prohibited) and *contrary conflicts* (a given behaviour is prescribed and prohibited). Following this definition, consistency of security policies has then been studied in [5].

If consistency of policies is a notion that has been rather well studied, completeness has, at the opposite, received much less attention. [2] proposes a definition of completeness between two confidentiality policies (for each piece of information, the user must have either the permission to know it or the prohibition to know it), definition which has been adapted in [7] for multilevel security policies.

Recently, focusing on information exchange policies, a definition of consistency and a definition of completeness have been given in [6]. These definitions have constituted a starting point for the present work and have been refined.

This paper is organised as follows.

Section 2 presents the logical formalism used to express information exchange policies, the definition of consistency of such policies as well as the definition we give of completeness. Section 3 focuses on the problem of reasoning with an incomplete policy. Following the approach that has led to the CWA (Closed World Assumption) in Database area [14], we will present some rules of completion that can be used in order to complete an incomplete policy. Then, following [12], we will define some default rules and we will prove the equivalence between these two solutions. Section 4 is devoted to a discussion and extensions of this work will be mentioned.

## 2 Information Exchange Policies

### 2.1 Preliminaries

We use the framework defined in [6] to represent a sharing policy. This logical framework,  $L$ , is based upon a typed first order logic<sup>1</sup>. The alphabet of  $L$  will

<sup>1</sup> We use a first order logic instead of a modal deontic logic mainly because imbricating deontic modalities is not needed here. Furthermore, this allows us to use the results on policies consistency provided in [3].

be based on four distinct groups of symbols: constant symbols, variable symbols, predicate symbols and function symbols. As we want to type the language, we will distinguish different groups of symbols among those four categories.

**Definition 1.** We distinguish three sets of constants: *ag-constants* (constants for agents), *i-constants* (constants for pieces of information), *o-constants* (other constants) and we distinguish three sets of variables: *ag-variables* (variables for agents), *i-variables* (variables for pieces of information), *o-variables* (other variables).

**Definition 2.** Predicate symbols are:

- *D-predicates*: unary predicates  $O$ ,  $P$ ,  $F$  and  $T$  (meaning respectively *Obligatory*, *Permitted*, *Forbidden*, *Tolerated*).
- *P-predicates*: predicates used to express any kind of property on pieces of information, agents, etc.

**Definition 3.** Functions symbols are:

- *i-functions*: used to represent properties about the pieces of information.
- $\text{not}(\cdot)$ : unary-function used to represent object level negation.
- $\text{tell}(\cdot, \cdot, \cdot)$ : function with three arguments representing the action of telling a piece of information.  $\text{tell}(x, y, i)$  represents the event created by an agent  $x$  making the action of telling  $y$  a piece of information  $i$ .

**Definition 4.** Terms are defined the following way :

- *ag-term* : ag-constant or ag-variable
- *i-term* : *i-constants* and *i-variables* are *i-terms*. If  $f$  is an *i-function* and  $i_1, \dots, i_n$  are *i-terms* then  $f(i_1, \dots, i_n)$  is an *i-term*.
- *d-term* : If  $x$  and  $y$  are ag-terms and  $i$  is an *i-term* then  $\text{tell}(x, y, i)$  is a *d-term*. Moreover, if  $d$  is a *d-term* then  $\text{not}(d)$  is a *d-term* too.
- *o-term* : *o-constant* or *o-variable*

**Definition 5.** Formulas of  $L$  are defined recursively as follows:

- Let  $d$  be a *d-term*. Then  $O(d)$ ,  $P(d)$ ,  $F(d)$  and  $T(d)$  are *D-literals* and formulas of  $L$ .
- If  $t_1, \dots, t_n$  are terms (other than *d-terms*) and  $P$  a *P-predicate* then  $P(t_1, \dots, t_n)$  is a *P-literal* and a formula of  $L$ .
- Let  $F_1$  and  $F_2$  be formulas of  $L$  and  $x$  be a variable. Then  $\neg F_1$ ,  $F_1 \wedge F_2$ ,  $F_1 \vee F_2$ ,  $\forall x F_1$ ,  $\exists x F_1$ ,  $F_1 \rightarrow F_2$  and  $F_1 \leftrightarrow F_2$  are formulas of  $L$ .

*Example 1.* We introduce here an example, that will be developed all along the paper. Let us consider the following logical language  $L$ :  $a, b, c, \text{Boss}$  and  $\text{Employee}$  are ag-constants and  $x$  and  $y$  are ag-variables. We can do the same for *i-terms*, etc.  $\text{Role}(\cdot, \cdot)$  is a *P-predicate*.  $\text{Role}(a, \text{Boss})$  means that agent  $x$  plays the role *Boss*.  $\text{Topic}(\cdot, \cdot)$  is a *P-predicate*.  $\text{Topic}(i_1, \text{ExpRisk})$  means that the piece of information  $i_1$  deals with topic *ExpRisk* (standing for *Explosion Risk*).  $\text{Agent}(\cdot)$  is a *P-predicate*.  $\text{Agent}(b)$  means that  $b$  is an agent.  $\text{Receive}(a, i_1)$  is a *L-literal* meaning that agent  $a$  receives the piece of information  $i_1$ .  $O(\text{tell}(x, y, i))$  is a *D-literal* meaning that agent  $x$  is obligated to tell agent  $y$  the piece of information  $i$ .

## 2.2 Information Exchange Policies

In this section, we define rules for an information sharing policy, within the above logical language.

**Definition 6.** *An information sharing policy is a set of formulas of  $L$  which are conjunction of clauses  $l_1 \vee l_2 \vee \dots \vee l_n$  such that:*

- $l_n$  is the only positive literal and is a  $D$ -literal,
- $\forall i \in \{1, \dots, n-1\}$ ,  $l_i$  is a negative  $L$ -literal,  $P$ -literal or  $D$ -literal,
- if  $x$  is a variable in  $l_n$ , then  $\exists i \in \{1, \dots, n-1\}$  such that  $l_i$  is a negative literal and contains the variable  $x$ .

*Example 2.* The rule "If a boss receives a piece of information dealing with the topic equipment checking, then it's forbidden for him to say it to his employees" is expressed with the following formula:

$$(R_0) \quad \forall(x, y, i) \quad \text{Role}(x, \text{Boss}) \wedge \text{Role}(y, \text{Employee}) \\ \wedge \text{Receive}(x, i) \wedge \text{Topic}(i, \text{EqChk}) \rightarrow F(\text{tell}(x, y, i))$$

## 2.3 Consistency and Completeness of Policies

We note  $\mathcal{A}$  the following set of axioms:

$$\begin{array}{ll} (Ax1) \quad \forall x \quad P(x) \leftrightarrow \neg O(\text{not}(x)) & (Ax2) \quad \forall x \quad F(x) \leftrightarrow O(\text{not}(x)) \\ (Ax3) \quad \forall x \quad T(x) \leftrightarrow P(x) \wedge P(\text{not}(x)) & (D) \quad \forall x \quad O(\text{not}(x)) \rightarrow \neg O(x) \\ (NO) \quad \forall x \quad O(\text{not}^{2n}(x)) \leftrightarrow O(x) & (NP) \quad \forall x \quad P(\text{not}^{2n}(x)) \leftrightarrow P(x) \\ (NF) \quad \forall x \quad F(\text{not}^{2n}(x)) \leftrightarrow F(x) \end{array}$$

NOTATION : Let  $A_1, A_2$ , and  $A_3$  be formulas of  $L$ . We will note:  $A_1 \otimes A_2$  instead of  $(A_1 \vee A_2) \wedge \neg(A_1 \wedge A_2)$  and  $A_1 \otimes A_2 \otimes A_3$  instead of  $(A_1 \vee A_2 \vee A_3) \wedge \neg(A_1 \wedge A_2) \wedge \neg(A_2 \wedge A_3) \wedge \neg(A_1 \wedge A_3)$ . This notation means that one and only one of the formulas  $A_i$  is true.

**Theorem 1.**  $\forall d$   $d$ -term  $\mathcal{A} \models O(d) \otimes T(d) \otimes F(d)$

*Proof.*  $\forall x, \neg T(x) \leftrightarrow_{Ax3} \neg P(x) \vee \neg P(\text{not}(x))$ . Yet,  $\neg P(x) \leftrightarrow_{Ax1} O(\text{not}(x)) \leftrightarrow_{Ax2} F(x)$  and  $\neg P(\text{not}(x)) \leftrightarrow_{Ax1} O(\text{not}^2(x)) \leftrightarrow_{NO} O(x)$ . Thus  $\mathcal{A} \models \neg T(x) \leftrightarrow O(x) \vee F(x)$  and  $\mathcal{A} \models O(x) \vee T(x) \vee F(x)$ . Then, we have  $\mathcal{A} \models \neg(O(d) \wedge T(d))$ ,  $\mathcal{A} \models \neg(O(d) \wedge F(d))$  and  $\mathcal{A} \models \neg(T(d) \wedge F(d))$ .

**Definition 7.** *A formula or a set of formulas  $S$  is complete if and only if, for all  $P$ -literal  $l$ , we have:  $S \models l$  or  $S \models \neg l$ .*

**Definition 8.** *A state of the world or a world,  $W$ , is a set of atomic formulas of  $L$  without  $D$ -literals. If this set is complete, we speak about a complete world.*

Let us note  $Dom$  the set of constraints that are supposed to be true in all worlds.

**Definition 9.** Let  $\mathcal{P}$  be a policy defined as a set of formulas of  $L$ , and  $W$  a complete world ruled by  $\mathcal{P}$ .  $\mathcal{P}$  is consistent in  $W$  (according to  $Dom$ ) if and only if  $(W \wedge Dom \wedge \mathcal{P} \wedge \mathcal{A})$  is consistent.

*Example 3.* We take  $Dom = \{\}$ ; Let us consider the following world  $W_0$ <sup>2</sup>:

$$W_0 = \{Agent(a), Agent(b), Role(a, Boss), Role(b, Employee) \\ Theme(i_1, EqtChk), Theme(i_2, ExpRisk), Receive(a, i_2)\}.$$

Let  $\mathcal{P}_0$  be a policy containing one rule which is the rule  $(R_0)$ .  $(W_0, Dom, \mathcal{P}_0, \mathcal{A})$  is consistent. Thus,  $\mathcal{P}_0$  is consistent in  $W_0$ .

**Definition 10.** Let  $\mathcal{P}$  be a policy.  $\mathcal{P}$  is consistent (according to  $Dom$ ) if and only if there is no set of formulas  $f$  of  $L$  without  $D$ -literal such that  $(f \wedge Dom)$  is consistent and  $(\mathcal{P} \wedge \mathcal{A} \wedge f \wedge Dom)$  is inconsistent.

**Proposition 1.**  $\mathcal{P}$  is consistent (according to  $Dom$ ) if and only if for all complete world  $W$ ,  $\mathcal{P}$  is consistent in  $W$ .

*Proof.* This can be proved by using contraposition.

*Example 4.* Let  $(R_1)$  be the following rule: "When an employee receives any piece of information about equipment check, it's tolerated for him to tell it to another employee".  $(R_1)$  can be formalized in the following way:

$$(R_1) \quad \forall(x, y, i) \quad Role(x, Employee) \wedge Role(y, Employee) \wedge \neg(x = y) \\ \wedge Receive(x, i) \wedge Topic(i, EqtChk) \rightarrow T(tell(x, y, i))$$

Let us consider the policy  $\mathcal{P}_1$  containing rules  $(R_0)$  and  $(R_1)$ . If we take  $f = Role(a, Employee) \wedge Role(a, Boss)$ , then  $(R_0)$  allows us to infer  $F(tell(a, y, i_2))$  and  $(R_1)$  to infer  $T(tell(a, y, i_2))$ . Thus, we have a contradiction and  $\mathcal{P}_1$  is not consistent in  $W_0$  so not globally consistent.

Intuitively, for a given world, a policy is complete if it allows to deduce the behaviour that any agent should have, according to any piece of information and according to any other agent he could tell this piece of information. It could be obligatory, forbidden or tolerated for the agent to say the piece of information to the other agent.

**Definition 11.** Let  $\mathcal{P}$  be a policy and  $W$  a complete world ruled by  $\mathcal{P}$ .  $\mathcal{P}$  is complete for  $\models$  in  $W$  if and only if, for all  $X = (x, y, i)$

$$\text{If } W \models Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \text{ Then}$$

$$(\mathcal{P}, W, \mathcal{A} \models O(tell(X)) \text{ or } \mathcal{P}, W, \mathcal{A} \models F(tell(X)) \text{ or } \mathcal{P}, W, \mathcal{A} \models T(tell(X)))$$

<sup>2</sup> We will write in  $W$  only the positive literals for more readability. Each literal that is not explicitly written in  $W$  will be considered as negative.

This definition can be generalized and we can define a global completeness.

**Definition 12.** Let  $\mathcal{P}$  be a policy.  $\mathcal{P}$  is globally complete for  $\models$  if and only if for all complete world  $W$ ,  $\mathcal{P}$  is complete for  $\models$  in  $W$ .

*Example 5.* We have  $W_0 \models \text{Receive}(a, i_2) \wedge \text{Agent}(b) \wedge \neg(a = b)$  but  $\mathcal{P}_0, W_0, \mathcal{A} \not\models O(\text{tell}(a, b, i_2))$  and  $\mathcal{P}_0, W_0, \mathcal{A} \not\models T(\text{tell}(a, b, i_2))$  and  $\mathcal{P}_0, W_0, \mathcal{A} \not\models F(\text{tell}(a, b, i_2))$ . Thus,  $\mathcal{P}_0$  is incomplete for  $\models$ .

Completeness is an important issue for a policy. For a given situation, without any behaviour stipulated, any behaviour could be observed and thus consequences could be quite important. With an incomplete policy, we could detect the "holes" of the policy and send them back to the policy designers so that they can correct them or we could detect the "holes" of the policy and allow for those holes some default rules that could be applied to correct them. The first solution could be quite irksome to be applied (the number of holes could be quite important and thus correct them one by one quite long). Then, we put in place the second solution.

### 3 Reasoning with Incomplete Policies

#### 3.1 Completion Rules

In this paragraph, we present a solution which extends the CWA defined by Reiter to complete first order databases.

According to CWA, if the database is incomplete for a literal  $l$  (i.e  $l$  is not deduced in the database), then it can be assumed that its negation ( $\neg l$ ) is deduced. This rule is motivated by the assumption that a database is used to represent the real world. Since in the real world, a fact is true or is false (i.e  $l \otimes \neg l$  is a tautology in first order logic) then a database must deduce a fact or its negation.

Here, given a d-term  $l$ , we are not interested in its truth value but in the fact that a given policy deduces that it is obligatory, forbidden or tolerated. These three cases are the only ones because axioms  $\mathcal{A}$  imply  $O(l) \otimes F(l) \otimes T(l)$ . Thus, if the policy is incomplete for a literal  $l$  (i.e it does not deduce neither  $O(l)$  nor  $F(l)$  nor  $T(l)$ ) then it can only be completed by assuming that  $O(l)$  can be deduced, or  $P(l)$  or  $F(l)$ . This leads to the three completion rules which are described in the following.

Furthermore, in order to be as general as possible, we define parametrized completion rules so that the way of completing by  $O(l)$ ,  $P(l)$  or  $F(l)$  may depend on some conditions. These conditions, denoted  $E_i$  in the following, will represent properties about agents (e.g, agents having a specific role), information (pieces of information dealing with a specific topic), etc.

Let  $\mathcal{P}$  be a consistent policy and  $W$  be a complete world ruled by  $\mathcal{P}$ .

**Notation.** For more readability, we will write " $\mathcal{P}, W$  incomplete for  $(x, y, i)$ " instead of:  $W \models \text{Receive}(x, i) \wedge \text{Agent}(y) \wedge \neg(x = y)$  and  $\mathcal{P}, W, \mathcal{A} \not\models O(\text{tell}(x, y, i))$  and  $\mathcal{P}, W, \mathcal{A} \not\models T(\text{tell}(x, y, i))$  and  $\mathcal{P}, W, \mathcal{A} \not\models F(\text{tell}(x, y, i))$ .

Let  $E_1$ ,  $E_2$  and  $E_3$  be three formulas that depend on  $x$  and/or on  $y$  and/or on  $i$ . We will write  $X$  instead of  $(x, y, i)$

The three inference rules are:

$$\begin{aligned} (R_{E_1}) \quad & \frac{\mathcal{P}, W \text{ incomplete for } X, \quad W \models E_1(X)}{F(\text{tell}(X))} \\ (R_{E_2}) \quad & \frac{\mathcal{P}, W \text{ incomplete for } X, \quad W \models E_2(X)}{T(\text{tell}(X))} \\ (R_{E_3}) \quad & \frac{\mathcal{P}, W \text{ incomplete for } X, \quad W \models E_3(X)}{O(\text{tell}(X))} \end{aligned}$$

We can complete an incomplete policy so that it is obligatory ( $R_{E_1}$ ), forbidden ( $R_{E_2}$ ) or tolerated ( $R_{E_3}$ ) for an agent to tell another agent a piece of information, according to those three elements. We define here a **new inference** that we will note  $\models_*$ . Rules of inference for  $\models_*$  are the same as for  $\models$  but we add  $R_{E_1}$ ,  $R_{E_2}$  and  $R_{E_3}$ .

The next step is to verify that the policy is complete and consistent with this new inference.

First of all, we have to extend the definition of completeness of a policy with the inference  $\models_*$ .

**Definition 13.** Let  $\mathcal{P}$  be a policy and  $W$  a complete world ruled by  $\mathcal{P}$ .  $\mathcal{P}$  is complete for  $\models_*$  in  $W$  if and only if for all  $X = (x, y, i)$ , we have:

$$\text{If } W \models \text{Receive}(x, i) \wedge \text{Agent}(y) \wedge \neg(x = y) \text{ Then}$$

$$(\mathcal{P}, W, \mathcal{A} \models_* O(\text{tell}(X)) \text{ or } \mathcal{P}, W, \mathcal{A} \models_* T(\text{tell}(X)) \text{ or } \mathcal{P}, W, \mathcal{A} \models_* F(\text{tell}(X)))$$

This definition can be generalized.

**Definition 14.** Let  $\mathcal{P}$  be a policy.  $\mathcal{P}$  is globally complete if and only if for all complete world  $W$ ,  $\mathcal{P}$  is complete in  $W$ .

**Proposition 2.** Let  $\mathcal{P}$  be a policy and  $W$  a complete world ruled by  $\mathcal{P}$ .  $\mathcal{P}$  is complete for  $\models_*$  in  $W$  if and only if

$$\forall X = (x, y, i), \mathcal{P}, W \text{ incomplete for } X \Rightarrow W \models E_1(X) \vee E_2(X) \vee E_3(X))$$

*Proof.* This can be proved by reasoning with contraposition.

*Example 6.*  $E_1(x, y, i) = \text{Topic}(i, \text{Eq}t\text{Check})$ ,  $E_2(x, y, i) = \text{False}$ ,  $E_3(x, y, i) = \text{Topic}(i, \text{ExpRisk})$ . We have  $\mathcal{P}_0, W_0$  incomplete only for  $(a, b, i_2)$ . We have  $W_0 \models E_3(a, b, i_2)$  so  $W_0 \models (E_1(a, b, i_2) \vee E_2(a, b, i_2) \vee E_3(a, b, i_2))$ . Then the policy  $\mathcal{P}_0$  is complete in  $W_0$ .

Then, we have to extend the definition of consistency for the new inference.

**Definition 15.** Let  $W$  be a complete a world and  $\mathcal{P}$  a policy that is consistent for  $\models$  in  $W$ <sup>3</sup>.  $\mathcal{P}$  is consistent for  $\models_*$  in  $W$  (according to domain  $Dom$ ) if and only if  $W, Dom, \mathcal{P}, \mathcal{A}$  is consistent for  $\models_*$  (i.e  $W, Dom, \mathcal{P}, \mathcal{A} \not\models \perp$ ).

**Proposition 3.** A policy  $\mathcal{P}$  that is complete for  $\models_*$  in a complete world  $W$  is consistent for  $\models_*$  in  $W$  (according to  $Dom$ ) if and only if

$\forall X = (x, y, i)$  If  $\mathcal{P}, W$  incomplete for  $(x, y, i)$  Then

$$W \models \neg(E_1(X) \wedge E_2(X)) \wedge \neg(E_1(X) \wedge E_3(X)) \wedge \neg(E_2(X) \wedge E_3(X))$$

*Proof.* This can be proved by reasoning with contraposition.

*Example 7.* We take  $E_1(x, y, i) = Topic(i, EqtChk)$ ,  $E_2(x, y, i) = False$  and  $E_3(x, y, i) = Topic(i, ExpRisk)$ . We have verified that  $\mathcal{P}_0$  is complete for  $\models_*$  in  $W_0$ .  $\mathcal{P}_0, W_0$  is incomplete for  $(a, b, i_2)$ . For this triplet, we have  $W_0 \models \neg(E_1(a, b, i_2) \wedge E_3(a, b, i_2))$ . Thus, the policy  $\mathcal{P}_0$  is consistent for  $\models_*$  in  $W_0$ .

**Corollary 1.** Let  $\mathcal{P}$  be a policy and  $W$  a world ruled by  $\mathcal{P}$ .  $\mathcal{P}$  is consistent and complete for  $\models_*$  in  $W$  if and only if

$\forall X = (x, y, i)$  If  $\mathcal{P}, W$  incomplete for  $(x, y, i)$  Then  $W \models E_1(X) \otimes E_2(X) \otimes E_3(X)$

**Definition 16.** A policy  $\mathcal{P}$  is globally consistent for  $\models_*$  (according to  $Dom$ ) if and only if it is consistent for  $\models_*$  in all complete world  $W$  where  $W \wedge Dom$  is consistent.

### 3.2 Default Rules

The three rules that we have just defined look similar to default logic defined by Reiter in [12]. The aim of this section is to develop this aspect. As a reminder to the default theory, one could read the chapter dedicated to default rules in [11]. Let  $\mathcal{P}$  be a policy in a complete world  $W$ . We suppose that  $\mathcal{P}$  is consistent in  $W$  for  $\models$ . Let  $W'$  be the set of formulas defined by  $W' = \mathcal{P} \cup W \cup \mathcal{A}$ . We define three default rules in  $W'$  and for that, we consider a triplet  $X = (x, y, i)$ .

$$(d_1) \frac{Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \wedge E_1(X) : F(tell(X))}{F(tell(X))}$$

$$(d_2) \frac{Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \wedge E_2(X) : T(tell(X))}{T(tell(X))}$$

$$(d_3) \frac{Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \wedge E_3(X) : O(tell(X))}{O(tell(X))}$$

The default rule  $d_1$  can be read as following: "If, in  $W'$ , an agent  $x$  receives a piece of information  $i$ , if  $y$  is another agent, if  $X = (x, y, i)$  are such that  $E_1(X)$

<sup>3</sup> It's not relevant to study a policy that is not consistent in  $W$ .



is true, and if it is consistent to suppose that it is forbidden for  $x$  to say  $i$  to  $y$ , then we consider that this forbidding is true in  $W'$ ".

We note  $D = \{d_1, d_2, d_3\}$ .  $d_j$  ( $j \in \{1, 2, 3\}$ ) is applicable if its prerequisite can be inferred in  $W'$  and if the negation of its justification cannot be inferred in  $W'$ . We consider now the theory  $\Delta = (D, W')$  and we look at its possible extensions.

**Proposition 4.** *The default theory  $\Delta = (D, W')$  has at least one consistent extension.*

*Proof.*  $\Delta = (D, W')$  is a closed normal default theory so we can use Reiter's theorem (theorem 3.1 in [12]) that says that "Every closed normal default theory has an extension". Thus, as  $W'$  is consistent, we can deduce that this extension is consistent.

*Example 8.* We build the three default rules with  $E_1(x, y, i) = Topic(i, EqChk)$ ,  $E_2(x, y, i) = False$  and  $E_3(x, y, i) = Topic(i, ExpRisk)$ . We have:

$W'_0 \models Receive(a, i_2) \wedge Agent(b) \wedge \neg(a = b) \wedge E_3(a, b, i_2)$  and  $W'_0 \not\models \neg O(tell(a, b, i_2))$ . The default rule  $d_3$  is then applicable for  $(a, b, i_2)$ . An extension of  $\Delta_0 = (D, W'_0)$  could be  $E_{\Delta_0} = Th(W'_0) \cup \{O(tell(a, b, i_2)), P(tell(a, b, i_2))\}$ .

We use here the universal inference for default rules : Let  $\varphi$  be a formula of  $L$ .  $W \models^{UNI, D} \varphi$  if and only if  $\varphi$  belongs to every extension of  $(D, W)$ .

**Proposition 5.** *Let  $\mathcal{P}$  be a policy applied in a complete world  $W$ . We suppose that  $\mathcal{P}$  is consistent in  $W$ .*

$\Delta = (D, W')$  has one and only one extension  $E_\Delta$  if and only if

$$\forall X = (x, y, i) \text{ If } \mathcal{P}, W \text{ incomplete for } X \text{ Then}$$

$$W \models \neg(E_1(X) \wedge E_2(X)) \wedge \neg(E_1(X) \wedge E_3(X)) \wedge \neg(E_2(X) \wedge E_3(X))$$

*Example 9.*  $\mathcal{P}, W$  is incomplete only for  $X_0 = (a, b, i_2)$ . We have  $W \models \neg(E_1(X_0) \wedge E_2(X_0)) \wedge \neg(E_1(X_0) \wedge E_3(X_0)) \wedge \neg(E_2(X_0) \wedge E_3(X_0))$  then  $E_{\Delta_0}$  is the only extension of  $\Delta_0$ .

**Definition 17.** *Let  $\mathcal{P}$  be a policy applied in a complete world  $W$ .  $\mathcal{P}$  is consistent for  $Dom$  in  $W$  for  $\models^{UNI, D}$  if and only if  $(D, W' \cup Dom)$  has one and only one extension.*

**Definition 18.** *Let  $\mathcal{P}$  be a policy applied in a complete world  $W$ .  $\mathcal{P}$  is complete for  $\models^{UNI, D}$  in  $W$  if and only if we have*

$$\forall X = (x, y, i) \text{ If } W \models Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \text{ Then}$$

$$(W' \models^{UNI, D} O(tell(X)) \text{ or } W' \models^{UNI, D} T(tell(X)) \text{ or } W' \models^{UNI, D} F(tell(X)))$$

**Proposition 6.** *Let  $\mathcal{P}$  be a policy applied in a complete world  $W$ .  $\mathcal{P}$  is consistent for  $Dom$  in  $W$  and is complete in  $W$  for  $\models^{UNI, D}$  if and only if*

$$\forall X = (x, y, i) \text{ If } \mathcal{P}, W \text{ incomplete for } X \text{ Then } W \models E_1(X) \otimes E_2(X) \otimes E_3(X)$$

*Example 10.* For  $X_0 = (a, b, i_2)$ , we have  $W_0 \models E_1(X_0) \otimes E_2(X_0) \otimes E_3(X_0)$ .  $P_0$  is consistent and complete in  $W_0$  pour  $\models^{UNI, D}$ .

### 3.3 Comparison

The two methods that we have just seen look very similar. The following proposition shows their relation.

**Proposition 7.** *Let  $W$  be a complete world and  $\mathcal{P}$  a policy that rules  $W$ . For all  $X = (x, y, i)$*

$$\begin{aligned}
 & \text{If } W \models \text{Receive}(x, i) \wedge \text{Agent}(y) \wedge \neg(x = y) \text{ Then} \\
 & (\mathcal{P}, W, \mathcal{A} \models_* F(\text{tell}(X)) \Leftrightarrow W' \models^{UNI,D} F(\text{tell}(X)) \wedge \\
 & \quad \mathcal{P}, W, \mathcal{A} \models_* T(\text{tell}(X)) \Leftrightarrow W' \models^{UNI,D} T(\text{tell}(X)) \wedge \\
 & \quad \mathcal{P}, W, \mathcal{A} \models_* O(\text{tell}(X)) \Leftrightarrow W' \models^{UNI,D} O(\text{tell}(X)))
 \end{aligned}$$

## 4 Discussion

After having given a logical framework and showed how to formalize a information exchange policy within this framework, we have reminded of a definition of consistency and we have defined what meant completeness for a policy. Therefore, the issue was to deal with incomplete policies. In solution to that, we have proposed two ways of completing a policy. One way is to use a new inference with three inference rules that can be applied for elements for which the policy is incomplete. The other one is to use the default theory and in particular three default rules that can be applied as soon as they are not in contradiction with what already exists. These default rules allow the construction of a new complete policy. After having completed policies, we can check that the result is still consistent. Finally, we prove that the methods results are equivalent. Indeed, in a given situation, as soon as an agent will receive a piece of information, the question will be to check if the policy deduces that it is obligatory, tolerated or forbidden for him to tell that information to another agent. If the answer is negative, then the question will be to check which condition  $I_i$  is true. If  $E_1$  (resp,  $E_2$ ,  $E_3$ ) is true, then the agent will deduce that it is forbidden (resp, tolerated, obligatory) to exchange this information. The condition on the  $I_i$ 's ensures that the agent will deduce one of them.

This work could be extended in many directions.

First, we could extend it by adding the notion of time. As it is shown in [10], this issue is very important when speaking about obligations. In our case, the impact of time will be quite difficult to deal with. Actually, we will have to consider different times : time the piece of information is created, time it is received by an agent, time the obligation is created, time the agent makes the action of telling the piece of information, time the obligation lasts.

Secondly, we could focus our attention on the so-called *Receive* predicate and study its semantic in relation with the agent's belief base revision. Indeed, the obligations, prohibitions, tolerances expressed by the policy should not be triggered by the arrival of a new piece of information in the agent's base, but by the computation of the "new" beliefs (i.e the ones which belong to the difference between the base before and after the revision).

Besides, it must be noticed that some properties other than consistency and completeness could be studied. For instance, we could wonder if the notion of *correctness*, as introduced in [8], is pertinent in our case. In that paper, the authors introduce the notion of *correctness* of airport security regulation. In their context, there are different organizations which do not have the same hierarchy level but which create rules on the same topics (e.g, dangerous objects on board a aircraft). The higher the hierarchy level of the organization is, the more general the rule is. The lower hierarchy level organizations have to create sub-rules that once all checked will validate the general rules. *Correctness* ensures that the sub-rules fulfill the general rules.

Finally, this present work could be extended to any kind of regulations. Indeed, it must be noticed that the idea which underlines the notion of completeness studied here for information exchange policies could be used to characterize a kind of "local completeness" (completeness relatively to some particular situations) for any type of regulation. This notion of "local completeness" is rather similar to the one already introduced in Database area. Indeed, as mentioned in [13] and [9], some integrity constraints expressed on a database are in fact rules about what the database should know. For instance, considering a database of a multi-national company, the integrity constraint "any employee has got a phone number, a fax number or a mail adress" expresses in fact that, for any employee known by the database, the database knows its phone number, its fax number or its mail address.<sup>4</sup> As first mentionned by Reiter [13], this integrity constraint expresses a kind of local completeness of the database. Defaults, as reiter defined them, can be used in order to complete such a database in case of incompleteness. For instance, one of the rules can be that if the database does contain any required information (no phone number, no fax number, no mail address) for a given employee but if the department that employee works in is known, then it can be assumed that its phone number is the phone number of its department.

## Acknowledgements

The authors wish to thank the reviewers for their helpful criticism.

## References

1. IEEE international conference on systems of systems engineering (2006)
2. Bieber, P., Cuppens, F.: Expression of confidentiality policies with deontic logic. In: Proceedings of the First Workshop on Deontic Logic and Computer Science (DEON'91) (1991)
3. Cholvy, L.: An application of SOL-deduction: checking regulation consistency. In: IJCAI'97 Poster Collection (1997)

---

<sup>4</sup> Notice that this does not prevent the fact that in the real world, an employee of the company has no telephone number, no fax number and no mail address.

4. Cholvy, L.: Checking regulation consistency by using SOL-resolution. In: International Conference on Artificial Intelligence and Law, pp. 73–79 (1999)
5. Cholvy, L., Cuppens, F.: Analysing consistency of security policies. In: IEEE Symposium on Security and Privacy (1997)
6. Cholvy, L., Garion, C., Saurel, C.: Modélisation de réglementations pour le partage d'informations dans un SMA. In: Modèles Formels de l'Interaction (2007)
7. Cuppens, F., Demolombe, R.: A modal logical framework for security policies. *Lectures Notes in Artificial Intelligence*, vol. 1325. Springer, Heidelberg (1997)
8. Delahaye, D., Etienne, J., Donzeau-Gouge, V.: Reasoning about airport security regulations using the focal environment. In: 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (2006)
9. Demolombe, R.: Database validity and completeness: another approach and its formalisation in modal logic. In: Proc. IJCAI Workshop on Knowledge representation meets databases (1999)
10. Demolombe, R., Bretier, P., Louis, V.: Formalisation de l'obligation de faire avec délais. In: Proc. MFI'2005 (2005)
11. LEASOMBE. Logique des défauts. In: Raisonement sur des informations incomplètes en intelligence artificielle. Teknea, Marseille (1989)
12. Reiter, R.: A logic for default reasoning. *Artificial Intelligence* 13(1,2) (1980)
13. Reiter, R.: What should a database know? *J. Log. Program.* 14(1-2), 127–153 (1992)
14. Reiter, R.: On closed world data bases. In: Nicolas, J.M., Gallaire, H., Minker, J. (eds.) *Logic and Databases*, Plenum Publications, New-York (1998)
15. Vranes, E.: The definition of "norm conflict" in international law and legal theory. *The European Journal of International Law* 17(2), 395–418 (2006)