Modeling Disjunctive Context in Access Control

Narhimene Boustia, Saad Dahlab
University of Blida, Algeria
nboustia@gmail.com

Aicha Mokhtari
USTHB, Algeria
aissani_mokhtari@yahoo.fr

***Abstract*—To provide dynamic authorizations to users, access control must take into account context. Using this idea, we develop a Contextual Multi-Level Access Control Model based on Description Logic with Default and Exception named $DL-CMLAC_{\delta\epsilon}$. To give a formal representation of this model, we define a non monotonic description logic based system by which we can deal with default and exceptional language called $JClassic^{+}_{\delta\epsilon}$. It is an extension of $JClassic_{\delta\epsilon}$ in order to introduce disjunction of concepts. $JClassic^{+}_{\delta\epsilon}$ is expressive enough to be of pratical use, it can handle a "weakened kind of disjunction" with the connective *lcs* allowing a tractable subsumption computation. The connective *lcs* has the same properties as the LCS external operation to compute the least common subsumer of two concepts. Connectives of $JClassic^{+}_{\delta\epsilon}$ are used in a cleaver way to represent authorization in a default context, an exceptional context and composed context.**

Keywords - Description logic; reasoner; disjunction; access control; context.

I. Introduction

The purpose of access control models is to assign permissions to users. The most interesting would be to have the ability to set dynamic permissions, i.e., context-dependent. Contexts express different types of extra conditions or constraints that control activation of rules expressed in the access control policy. there are several types of context:

- The Temporal context that depends on the time at which the subject is requesting for an access to the system,
- the Spatial context that depends on the subject location,
- the User-declared context that depends on the subject objective (or purpose),
- the Prerequisite context that depends on characteristics that join the subject, the action and the object.
- the Provisional context that depends on previous actions the subject has performed in the system.

We also assume that each organization manages some information system that stores and manages different types of information. To control context activation, each information system must provide the information required to check that conditions associated with the context definition are satisfied or not. The following list gives the kind of information related to the contexts we have just mentioned:

- A global clock to check the temporal context,
- the subject environment and the software and hardware architecture to check the spatial context,
- the subject purpose to check the user-declared context,
- the system database to check the prerequisite context,
- an history of the action carried out, to check the provisional context.

We are interested in modeling the user-declared context, particularly disjunction of several constraints [1].

Literature provides a wide range of access control models and policy languages. One of them is multilevel access control commonly used by military organisations in order to protect the confidentiality of their informations [2]. We propose to develop an access control model inspired from multilevel access control with the introduce of user-declared context to provide dynamic authorization.

In our model, authorization depends not only on classification and clearance levels, but also on the current context, in which user requests a right of access. Each change of context implies a change in permissions.

There are several type of user-declared contexts. It could be the regular one (what we call in our work default context) like it may be an exception to the current context. For example, in current days, each patient in a hospital is treated by his own doctor, but when there is an exception like an emergency, the authorization should change.

Context can be composed of several contexts (constraints). For example, under normal circumstances, the family has the right to visit the patient but when there is a risk of contamination and/or unknown disease, family loses this right. In this paper, we are interested in disjunction of context *(or)*.

To provide a formal representation, we use the $JClassic^{+}_{\delta\epsilon}$ [1] developed by us for this purpose. It is a description logic-based system augmented with two operators δ (for default) and ϵ (for exception) inspired by $AL_{\delta\epsilon}$ description logic [3] and "lcs" (for disjunction).

This kind of non-monotonic reasoning in description logic is not sufficiently developed. Actually, there is no system, in our sense, developed on this kind of reasoning in the web [4].

Description logics are powerful knowledge representation systems providing well-founded and computationally tractable classification reasoning. However, expression of disjunction of concepts has previously been infeasible due to computational cost.

Donini [5] shows that concept disjunction makes subsumption computation co-NP-Complete. However, disjunction is very useful for knowledge representation.

$JClassic^{+}_{\delta\epsilon}$ is an extension of $JClassic_{\delta\epsilon}$ [6], [7] by the operator of disjunction "lcs". This operator allows us to define context disjunction in access control.

The "lcs" connective has the same properties as the LCS

external (External insofar as LCS is not a connective of the language) operation introduced by Borgida et al. [8], which computes the least common subsumer of two concepts (The Least Common Subsumer of two concept A and B belonging to a language *L* is the most specific concept in *L* that subsumes both A and B) [9]. It was introduced by Ventos et al. in Classic to allow disjunction with a reasonable computation [10], [11].

Because of $JClassic_{\delta\epsilon}$ has been given an intensional semantics, $JClassic^{+}_{\delta\epsilon}$ is provided with an intentional semantics (called $\mathcal{CL}^{+}_{\delta\epsilon}$) based on an algebraic approach. For this, we have first to build an equational system, which highlights the main properties of the connectives. The equational system allows to define axiomatically the notion of LCS operation.

The main operation, the computation of the subsumption relation of $JClassic^{+}_{\delta\epsilon}$, is used to classify and deduce knowledge. The inheritance relation is used to compute the inherited properties.

In this paper, we first present our description logic system $JClassic^{+}_{\delta\epsilon}$ and we give definition of "lcs". We illustrate then the use of this reasoner for access control, using a small demonstration to show how authorization can be given to user.

II. Related work

Several approaches have been proposed to model context in access control. As we have seen earlier, the context makes it possible to express different kinds of constraint.

In RBAC family models, many works were done in this sense, some of them extend RBAC model to deal with access control based on user's location context [12], [13], [14], [15], [16], [17], or temporal context [18]. They suggest to combine concept of role with spatial or temporal condition to obtain contextual roles.

Georgiadis and al. [19] present a team- based access control model that is aware of contextual information associated with activities in application. Hu et al. [20] developed a context-aware access control model for distibuted healthcare application. To provide context-aware access control, the model defines the notion of context type and context constraint.

In OrBAC family, context is represented by an argument in the predicate ***Permission*** [21].

Several extension where done to take into account various types of context such as spatial, temporal and composed context [22], [23].

In these approaches, context is still modeled by an argument in a predicate using some algebra to write context, the authors don't present how the final value of context is calculated. Add to this, first order logic is known to be semi-decidable.

In our approach, context can be atomic or composed context using conjunction or disjunction.

III. $JClassic^{+}_{\delta\epsilon}$

$JClassic^{+}_{\delta\epsilon}$ is an non monotonic reasoner based on description logic with default and exception [3], which allows us to deal with default and exceptional knowledge.

The set of connectives of $JClassic^{+}_{\delta\epsilon}$ is the union of the set of connectives of $\mathcal{AL}_{\delta\epsilon}$ [3] presented in [7], [24], [25] and the connective "lcs".

The connective δ intuitively represents the common notion of default. For instance, having Animal as a conjunction with the concept δFly in the definition of the concept **Bird** states that birds generally fly.

The connective ϵ is used to represent a property that is not present in the description of the concept or of the instance but that should be. For instance, the definition of **Penguin** in $JClassic^{+}_{\delta\epsilon}$ is $Penguin \sqsubseteq Bird \sqcap Fly^{\epsilon}$. The Fly^{ϵ} property expresses the fact that fly should be in the definition of Penguin since it is a bird. The presence of Fly^{ϵ} in the definition of Penguin makes it possible to classify Penguin under the concept Bird.

Formally, the subsumption relation uses an algebraic semantics. The main interest of this approach is the introduction of the definitional point of view of default knowledge: from the definitional point of view, default knowledge can be part of concept definition whereas from the inheritance is only considered as a weak implication. A map between the definition of concept and its inherited properties is done with the calcuation of its normal form. This combinating of definitional and inheritance levels improves the classification process.

In this section, we first present the syntax of our system, we then give details about its algebraic semantic.

A. Syntax of $JClassic^{+}_{\delta\epsilon}$

The set of connectives of $JClassic^{+}_{\delta\epsilon}$ is the union of the set of connectives of $\mathcal{CL}_{\delta\epsilon}$ [6] and the connective *lcs*. $JClassic^{+}_{\delta\epsilon}$ is defined using a set **R** of primitive roles, a set **P** of primitive concepts, the constant $\bot$ (Bottom) and $\top$ (Top) and the following syntax rule (C and D are concepts, P is a primitive concept, R is a primitive role).

δ and ϵ are unary connectives, $\sqcap$ is a binary conjunction connective and $\forall$ enables universal quantification on role values. The Terminological language is given in Table 1.

B. Semantic of $JClassic^{+}_{\delta\epsilon}$

We endow $JClassic^{+}_{\delta\epsilon}$ with an intentional algebraic semantic denoted $\mathcal{CL}^{+}_{\delta\epsilon}$.

This framework covers the different aspects of the formal definition of concepts and subsumption in our language. The calculating of denotations of concepts in $\mathcal{CL}^{+}_{\delta\epsilon}$ is used in computing subsumption in the algorithm $Sub^{+}_{\delta\epsilon}$. $\mathcal{CL}^{+}_{\delta\epsilon}$ allows first to show that $Sub^{+}_{\delta\epsilon}$ is correct and complete and secondly to give a formal characterization of calculation of subsumption used in the implementation of $JClassic^{+}_{\delta\epsilon}$.

Subsumption is considered from two points of view:

- A descriptive point of view: it consists on the comparison of terms through an equational system;
- A structural point of view: it consists on a comparison of normal forms of concept

1) EQ: an equational system for $JClassic^{+}_{\delta\epsilon}$: In order to serve as the basis for the definition of an algebraic semantics, an equational system EQ is defined. From a descriptive point of view, the calculation of subsumption consists on the comparison of terms through the equational system **EQ**. This

$C, D \rightarrow \top$	the most general concept
$\mid \bot$	the most specific concept
$\mid P$	primitive concept
$\mid C \sqcap D$	concept conjunction
$\mid \neg P$	negation of primitive concept (This restriction to primitive concept in the negation is a choice to avoid the untractability)
$\mid \forall r : C$	C is a value restriction on all roles R
$\mid$ R AT-LEAST n	cardinality for R (minimum)
$\mid$ R AT-MOST n	cardinality for R (maximum)
$\mid \delta C$	default concept
$\mid C^{\epsilon}$	exception to the concept
$\mid ClcsD$	concept disjunction

TABLE I
SYNTAX OF $JClassic^{+}_{\delta\epsilon}$

system fixes the main properties of the connectives and is used to define an equivalence relation between terms and then to formalize the subsumption relationship.

$\forall A, B, C \in JClassic^{+}_{\delta\epsilon}$:
01: $(A \sqcap B) \sqcap C = A \sqcap (B \sqcap C)$
02: $A \sqcap B = B \sqcap A$
03: $A \sqcap A = A$
04: $\top \sqcap A = A$
05: $\bot \sqcap A = \bot$
06: $(\forall R : A) \sqcap (\forall R : B) = \forall R : (A \sqcap B)$
07: $\forall R : \top = \top$
08: R AT-LEAST m $\sqcap$ R AT-LEAST n = R AT-LEAST maxi(m,n)
09: R AT-LEAST 0 = $\top$
10: R AT-MOST m $\sqcap$ R AT-MOST n = R AT-MOST mini(m,n)
11: R AT-MOST 0 = $\forall R : \bot$
12: R AT-LEAST m $\sqcap$ R AT-MOST n (if n $\leq$ m).
13: (A lcs B) lcs C = A lcs (B lcs C)
14: A lcs B = B lcs A
15: A lcs A = A
16: A lcs $\top$ = $\top$
17: A lcs $\bot$ = A
18: $(\delta A)^{\epsilon} = A^{\epsilon}$
19: $\delta(A \sqcap B) = (\delta A) \sqcap (\delta B)$
20: $A \sqcap \delta A = A$
21: $A^{\epsilon} \sqcap \delta A = A^{\epsilon}$
22: $\delta\delta A = \delta A$
23: $(A^{\epsilon})^{\epsilon} = \delta A$

Axioms 01 to 12 are classical; they concern description logic connectives properties [26], [27]. Axioms 13 to 17 concern the connective "lcs". The following ones correspond to $\mathcal{AL}_{\delta\epsilon}$ connectives properties[3], i.e., properties of δ and ϵ connectives.

Descriptive Subsumption:

We denote $\sqsubseteq_d$ for descriptive subsumption. $\sqsubseteq_d$ is a partial order relation on terms. Equality (modulo the axioms of EQ) between two terms is denoted $=_{EQ}$. $=_{EQ}$ is a congruence relation which partitions the set of terms, i.e., $=_{EQ}$ allows to form equivalence classes between terms. We define the descriptive subsumption using the congruence relation and conjunction of concepts as follow:

Definition 1: (Descriptive Subsumption)
Let C and D two terms of $JClassic^{+}_{\delta\epsilon}$, $C \sqsubseteq_d D$, i.e., D subsume descriptively C, iff $C \sqcap D =_{EQ} C$.

From an algorithmic point of view, terms are not easily manipulated through subsumption. We adopt a structural point of view closer to the algorithmic aspect of computing subsumption. This allows us to first formalize calculation of subsumption in the implementation of $JClassic^{+}_{\delta\epsilon}$ and secondly to endow $JClassic^{+}_{\delta\epsilon}$ with an intensional semantics.

To define the subsumption relation between two concepts using their description, we need to compare them. For this, concepts are characterized by a normal form of their properties rather than by the set of their instances.

2) Normal Form of concept: We present in this section the structural point of view for the subsumption in $JClassic^{+}_{\delta\epsilon}$. This point of view has two main advantages: it is very close to the algorithmic aspects and is a formal framework to validate the algorithmic approach, which is not the case description graph.

We define a structural concept algebra $\mathcal{CL}^{+}_{\delta\epsilon}$, which is used to give an intensional semantic in which concepts are denoted by the normal form of their set of properties. The structural point of view of subsumption consist then to compare the normal forms derived by applying a homomorphism from set of terms of $JClassic^{+}_{\delta\epsilon}$ to elements of $\mathcal{CL}^{+}_{\delta\epsilon}$.

$\mathcal{CL}^{+}_{\delta\epsilon}$: an intensional semantic for $JClassic^{+}_{\delta\epsilon}$

From the class of CL-algebra, we present a structural algebra $\mathcal{CL}^{+}_{\delta\epsilon}$, which allows to endow $JClassic^{+}_{\delta\epsilon}$ with an intentional semantic.

Element of $\mathcal{CL}^{+}_{\delta\epsilon}$ are the canonical intentional representation of terms of $JClassic^{+}_{\delta\epsilon}$ (i.e., Normal form of the set of their properties). We call an element of $\mathcal{CL}^{=}_{\delta\epsilon}$ normal forms.

Definition of $\mathcal{CL}^{+}_{\delta\epsilon}$ means definition of a homomorphism h, which allows to associate an element of $\mathcal{CL}^{+}_{\delta\epsilon}$ to a term of $JClassic^{+}_{\delta\epsilon}$.

Using the equational system, we calculate for each concept

a structural denotation, which is a single normal form of this concept. The calculation of a normal form from a description of a concept can be seen as a result of term "rewriting" based on the equational system EQ.

The normal form of a concept defined with description T (noted nf(T)) is a pair $\langle t_\theta, t_\delta \rangle$ where t_θ contains strict properties of T and t_δ the default properties of T.

t_θ and t_δ are 6-uplet of the form (dom,min,max,π,r,ϵ) with:

dom: is a set of Individuals, if the description contain the property ONE-OF else the symbol UNIV.

min (resp. max): is a real, if the description contain the property Min (resp. Max) else the symbol MIN-R (resp. MAX-R).

π: is a set of primitive concept in description T.

r: have the form $\langle R, fillers, least, most, c \rangle$ where:

R: the name of Role.

fillers: set of Individuals, if the description contain the property R Fills, else $\emptyset$.

least (resp. most): is an integer, if the description contain AT-LEAST (resp. AT-MOST), else 0 (resp. NOLIMIT).

c: is the normal form of C, if the description contain the property $\forall R : C$.

ϵ: set of 6-uplet with the form (dom, min, max, π, r, ϵ).

Example: The normal form of concept $C \equiv A \sqcap \delta B$ is:

fn (C) = ($\langle Univ, Min-R, Max-R, \{A\}, \emptyset, \emptyset \rangle$, $\langle Univ, Min-R, Max-R, \{A, B\}, \emptyset, \emptyset \rangle$).

The interpretation of connectors and constants of $\mathcal{CL}^+_{\delta\epsilon}$ is given in Table 2. b_0 is a constant used as a denotation of $\bot$ [6].

Structural Subsumption:

Two terms C and D of $JClassic^+_{\delta\epsilon}$ are structurally equivalent iff their normal forms are equal. We denote $\sqsubseteq_s$ for structural subsumption. $\sqsubseteq_s$ is a partial order relation.

The structural equality of two terms of $JClassic^+_{\delta\epsilon}$ is noted $=_{CL}$. $=_{CL}$ is a congruence relation as $=_{EQ}$ in descriptive subsumption.

We define the structural subsumption using the congruence relation and conjunction of concepts as follow:

Definition 2: **(Structural Subsumption)**

Let C and D two terms of $JClassic^+_{\delta\epsilon}$, $C \sqsubseteq_s D$; i.e., D subsume structurally C, iff $C \sqcap D =_{CL} C$.

Theorem 1: **(Equivalency between descriptive subsumption and structural subsumption)**

Let C and D two terms of $JClassic^+_{\delta\epsilon}$, $C \sqsubseteq_s D \Leftrightarrow C \sqsubseteq_d D$.

To infer new knowledge in this system, the susbsumption relation is the main operation. In the next section, we outline the subsumption algorithm handling defaults and exceptions named $\text{Sub}_{\delta\epsilon}$.

IV. Inference in $JClassic^+_{\delta\epsilon}$

There are several reasoning services to be provided by a DL- system. We concentrate our work on the following basic ones, which are classification of concepts (TBox) and instance checking (ABox). These two services basically use the subsumption relation.

A. The Subsumption Relation

Borgida [8] defines the subsumption based on a set theoretic interpretation as follow: "The concept C subsume D, if and only if the set of instances of C include or is equal to a set of instances of D".

However, the general principle of computing subsumption between two concepts is to compare their sets of properties, not their sets of instances.

For this, we use an intensional semantics which is closer to the algorithmic aspects of computing subsumption, and this by defining a normal form of description called descriptive normal form.

Algorithm of Computing Subsumption $Sub^+_{\delta\epsilon}$

$Sub^+_{\delta\epsilon}$ is an algorithm of computing subsumption of the form Normalization- Comparison. It is consists of two steps, first, the normalization of description, and then a syntactic comparison of the obtained normal forms.

Let C and D be two terms of $JClassic^+_{\delta\epsilon}$. To answer the question "Is C subsumed by D?" we apply the following procedure. The normal forms of C and "C $\sqcap$ D" are calculated with the procedure of normalisation.

There are two steps in the comparison. We compare the strict parts of the two concepts. If these are equal, then we compare the default parts. If the two normal forms are equal, the algorithm returns "Yes". It returns "No" otherwise.

Algorithm 1 Algorithm $\text{Sub}^+_{\delta\epsilon}$

Require: C and D two description of concepts of $JClassic^+_{\delta\epsilon}$
Ensure: Response "Yes" or "No" to question "Is C subsumed by D?"

```
{Compute normal forms}
fn(C) ← Normalization(C)
fn(C ⊓ D) ← Normalization(C ⊓ D)
{Treatment of bottom}
if fn(C)=b0 then
  Response ← "Yes"
else
  if fn(C ⊓ D)=b0 then
    Response ← "No"
  else
    {Comparison of the obtained normal forms}
    Compar(fn(C)θ, fn(C⊓ D)θ, rep1)
    if rep1="Yes" then
      Compar(fn(C)δ, fn(C⊓ D)δ, rep1)
      Response ← rep2
    else
      Response ← "No"
    end if
  end if
end if
```

$JClassic_{\delta\epsilon}$	$\mathcal{CL}^{+}_{\delta\epsilon}$
$\top$	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$)$\succ$
P	$\prec$ (UNIV, MIN-R, MAX-R, P, $\emptyset$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$)$\succ$
ONE-OF E	$\prec$ (E, MIN-R, MAX-R, P, $\emptyset$, $\emptyset$), (E, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$)$\succ$
MIN u	$\prec$ (UNIV, u, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$), (UNIV, u, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$)$\succ$
MAX u	$\prec$ (UNIV, MIN-R, u, $\emptyset$, $\emptyset$, $\emptyset$), (UNIV, MIN-R, u, $\emptyset$, $\emptyset$, $\emptyset$)$\succ$
$\forall R : C(C \not\equiv \top etC \not\equiv \bot)$	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, 0, \lvert c_{\theta.dom}\rvert, c\rangle\}$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, 0, \lvert c_{\theta.dom}\rvert, c\rangle\}$, $\emptyset$)$\succ$
$\forall R : CetC \equiv \bot$	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, 0, 0, b_0\rangle\}$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, 0, 0, b_0\rangle\}$, $\emptyset$)$\succ$
$\forall R : CetC \equiv \top$	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$)$\succ$
$RFILLSE(E \neq \emptyset)$	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, E, \lvert E\rvert, NOLIMIT, t\rangle\}$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, E, \lvert E\rvert, NOLIMIT, t\rangle\}$, $\emptyset$)$\succ$
R FILLS $\emptyset$	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$)$\succ$
$RAT - LEASTn(n \neq 0)$	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, n, NOLIMIT, t\rangle\}$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, n, NOLIMIT, t\rangle\}$, $\emptyset$)$\succ$
R AT-LEAST 0	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$)$\succ$
$RAT - MOSTn(n > 0)$	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, 0, n, t\rangle\}$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, 0, n, t\rangle\}$, $\emptyset$)$\succ$
R AT-MOST 0	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, 0, 0, b_0\rangle\}$, $\emptyset$), (UNIV, MIN-R, MAX-R, $\emptyset$, $\{\langle R, \emptyset, 0, 0, b_0\rangle\}$, $\emptyset$)$\succ$
$C \sqcap D$	c $\otimes$ d
C lcs D	c LCS d
δ C	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, $\emptyset$), c_δ $\succ$
C^ϵ	$\prec$ (UNIV, MIN-R, MAX-R, $\emptyset$, $\emptyset$, c_δ), ($c_{\delta.dom}$, $c_{\delta.max}$, $c_{\delta.min}$, $c_{\delta\pi}$, $c_{\delta r}$, $c_{\delta\epsilon} \cup c_\delta$)$\succ$
$\bot$	b_0

TABLE II
INTERPRETATION OF CONNECTORS AND CONSTANTS OF $\mathcal{CL}^{+}_{\delta\epsilon}$

The completeness, correctenness and the polynomial computation of $JClassic_{\delta\epsilon}$ have been proved in [7].

B. Classification of concept

The classification of concepts is an operation which inserts a concept to the most appropriate place in the hierarchy. The classification process allows to find subsumption relations between concepts in the taxonomy (hierarchy) and insert new concept in the hierarchy.

In the $JClassic^{+}_{\delta\epsilon}$ reasoner, the classification of a concept consists of two phases: first phase is to find the most specific concepts that subsume the concept C (the concept C is to classify), they are called subsumeurs (SPS). The second phase search the most general concepts than C, we call them subsumed (GSP), it also establish new relations between the concept to classify C, its SPS and its GSP.

The classification process is triggered when we create a new concept (primitive or defined).

C. Instance recognition

The recognition of instances is to find for a given individual the most specific concepts which it may be an instance.

We used the method to achieve Abstraction-Classification mechanism instantiation of concepts.

The method Abstraction - Classification

This method allows the instantiation of the individual, it consists of two phases:

Abstraction: calculates the abstract concept of the individual containing all the information in the form of an abstract defined concept.

Classification: is to find the abstract concept of SPS, the SPS corresponds to the direct instances of individual, in other words: To determine whether an object O is instance of a concept C, we calculate the abstract concept AO, we then check if C subsumes AO. If so, we deduce that O is an instance of concept C, else O is not an instance of concept C.

Ex: if an individual named "Sara" eats only plants, the reasoner determine that Sara is an instance of concept VEGETARIAN.

D. Inheritance relation

The inheritance relation allows to compute the inherited properties of a concept. These properties are the basic ones in inferential systems.

The inheritance relation serves as a basis for retrieving the inherited properties, it also helps in distinguishing strict and

default inherited properties and answering questions concerning conflicts and consistency.

The main task of the inheritance relation is to retract exceptions from the denotation of a concept (the two ϵ parts of denotation). The inheritance-form scenario of a concept C is:

1- Replace each exception at an even level by a default in the denotation of C.

2- For each role of C, recursively call inheritance with the role value restriction.

3- Suppress P (resp. P°) in $c_{\delta\pi}$ if P° (resp. P) is in $c_{\phi\pi}$. The resulting denotation is called the inheritance form of C (The set of primitive concepts P is complemented with a new set P° in order to denote negation. There is no axiom that relates a primitive concept to its negation. This set P° is then theoretically necessary, but transparent for the user).

Using this relation, we deduce the inherited properties, and type them between strict and default ones. The inherited properties are those found in the inheritance form of the concept. The strict ones (resp. the default ones) are those in the strict (resp. default) part.

Algorithm 2 Inheritance Map

inheritance: $\mathcal{CL}^{+}_{\delta\epsilon} \rightarrow \mathcal{CL}^{+}_{\delta\epsilon}$, such that
inheritance(a)=
res $\leftarrow \prec$ ($a_{\theta\pi}$, $\emptyset$, $\emptyset$), ($a_{\delta\pi}$, $\emptyset$, $\emptyset$)$\succ$
for all y $\in a_{\theta\epsilon} \cup a_{\delta\epsilon}$ **do**
  res $\leftarrow$ res $\cup$ transform(y,$a_{\theta\epsilon}$)
end for
for all $\prec$r, p $\succ \in a_{\theta r}$ **do**
  res $\leftarrow$ res $\cup \prec$ ($\emptyset$, $\prec$ r, inheritance(p)$\succ$, $\emptyset$), ($\emptyset$, $\emptyset$, $\emptyset$)$\succ$
end for
for all $\prec$r, p $\succ \in a_{\delta r}$ **do**
  res $\leftarrow$ res $\cup \prec$ ($\emptyset$, $\emptyset$, $\emptyset$), ($\emptyset$, $\prec$ r, inheritance(p)$\succ$, $\emptyset$)$\succ$
end for
let res be $\prec$ ($res_{\theta\pi}$, $res_{\theta r}$, $res_{\theta\epsilon}$), ($res_{\delta\pi}$, $res_{\delta r}$, $res_{\delta\epsilon}$)$\succ$
for all x $\in res_{\theta\pi}$ **do**
  suppress x° from $res_{\delta\pi}$
end for
for all x° $\in res_{\theta\pi}$ **do**
  suppress x from $res_{\delta\pi}$
end for
return res

We detailed in the next section the connective *"lcs"* and the operation *LCS*, by which we compute normal form of ***A lcs B*** (A and B are concepts).

V. The computation of "lcs"

The least common subsumer has been introduced in description logic by Borgida et al. [8] as an external operation to compute the LCS of two concepts.

The LCS of two concepts A and B belonging to a language L is the most specific concept in L that subsumes both A and B.

Definition 3: Let L a terminological language, $\sqsubseteq$ the notation of subsumption relation in L

LCS: L $\times$ L $\rightarrow$ L

LCS(A,B) $\rightarrow$ C $\in$ L iff:

$A \sqsubseteq C$ and $B \sqsubseteq C$ (C subsume both A and B),

$\nexists$ D $\in$ L such that $A \sqsubseteq D, B \sqsubseteq D$ and $D \subseteq C$ (i.e., there is no common subsumers to A and B, which is subsumed strictly by C)

The next algorithm is to compute the LCS where input are the normal form of two concepts A_1 and A_2 and the output is the LCS of A_1 and A_2.

Let a and b two normal forms A and B with a and $b \neq b_0$ (b_0 is the normal form of $\perp$).

Algorithm 3 LCS

Require: a=$\prec a_\theta$,$a_\theta \succ$ and b=$\prec b_\theta$,$b_\theta \succ$ two normal forms of A and B.
Ensure: c=$\prec c_\theta$,$c_\theta \succ$ the normal form of LCS(A,B)
$c_{\theta\pi} \leftarrow a_{\theta\pi} \cap b_{\theta\pi}$
$c_{\theta r} \leftarrow \emptyset$
for all $\prec$r, d $\succ \in a_{\theta r}$ **do**
  if $\exists \prec$r, e $\succ \in b_{\theta r}$ **then**
    f $\leftarrow$ LCS(d,e)
    $c_{\theta r} \leftarrow c_{\theta r} \cup \prec$r, f $\succ$
  end if
end for
$c_{\theta\epsilon} \leftarrow a_{\theta\epsilon} \cap b_{\theta\epsilon}$
$c_{\delta\pi} \leftarrow a_{\delta\pi} \cap b_{\delta\pi}$
$c_{\delta r} \leftarrow \emptyset$
for all $\prec$r, d $\succ \in a_{\delta r}$ **do**
  if $\exists \prec$r, e $\succ \in b_{\delta r}$ **then**
    f $\leftarrow$ LCS(d,e)
    $c_{\delta r} \leftarrow c_{\delta r} \cup \prec$r, f $\succ$
  end if
end for
$c_{\delta\epsilon} \leftarrow a_{\delta\epsilon} \cap b_{\delta\epsilon}$

$JClassic^{+}_{\delta\epsilon}$ can be used in differents application. We will use it to formalize our contextual multilevel access control model, in which the context could be in different forms.

VI. Application to Access Control

To show how we can use our description logic-based system and how we can infer new knowledge, we define a knowledge base adapted to formalize a dynamic access control model named $DL - CMLAC_{\delta\epsilon}$ (Contextual Multi-Level Access Control Model based on Description Logic with Default and Exception).

In this model, authorization to subject is assigned depending on context. We consider first that the context is by default normal, and we represent it using the operator of default (δ). Then, each change of context is considered as an exception to the current context, this change is represented by the operator of exception (ϵ). We give, as an example, one ABox to show how authorization can be deduced.

We are intersted in one kind of policy, which is a multilevel access control commonly used by military organizations in order to protect the confidentiality of their informations [2].

In multilevel access control model, a subject *s* can access to an object *o* only if its clearance level is greater than or equal to the classification level of the object.

To allow the policy designer to define a security policy independently of the implementation, we introduce an abstract level.

Subject and Object are respectively abstracted into Role and View. A role is a set of subjects to which the same security rule apply and similarly, a view is a set of objects to which the same security rule apply. For example, the subject "John" plays the role of "Doctor" in the organization "Service of Pediatrics" and the view "Medical record" corresponds to the object "Medical record of patient". A clearance level is assigned to the role and a classification level is assigned to a View. An abstract authorization is assigned to a role on a view in a given context if its clearance level is greater than or equal to the classification level of the view.

The concrete authorization is derived from the abstract one depending on context.

In our approach, we take into account the context by considering the following postulate.

Postulate 1 (Normal context). *By default, the context is normal (usual context).*

Postulate 2. *All actions that are not permitted are prohibited.*

Figure 1 describes the general architecture of our access control model.

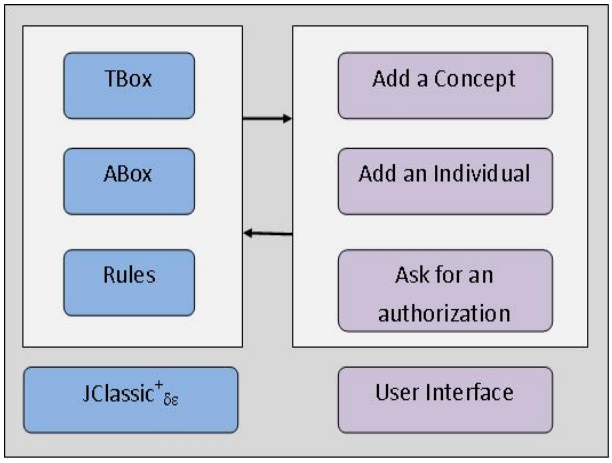

Fig. 1. Architecture of $DL - CMLAC_{\delta\epsilon}$

A. TBox

We now conceptualize access control model by a *DL* knowledge base capturing its characteristics, including the context with the use of defaults (δ) and exceptions (ϵ), using the Bell and La Padula model [28]. We define then TBox and ABox axioms with examples to illustrate their content and use.

We define a *DL* knowledge base $\mathcal{K}$. The alphabet of $\mathcal{K}$ includes the following atomic concepts: Subject, Object, Role, View, LevelR and LevelV. The TBox includes the following axioms.

- **Role attribution axiom:**
  $Subject \sqsubseteq \top$
  $Role \sqsubseteq \top$
  $Employ \sqsubseteq EmployS.Subject \sqcap EmployR.Role$
  It defines the relationship between subject and role, where:
  EmployS and EmployR are binary relations such as:
  - EmployS : EmployS links the concept Employ to the concept Subject
  - EmployR : EmployR links the concept Employ to the concept Role.

- **View definition axiom:**
  $Object \sqsubseteq \top$
  $View \sqsubseteq \top$
  $Use \sqsubseteq UseO.Object \sqcap UseV.View$
  It defines relationship between object and view, where:
  UseO and UseV are binary relations such as:
  - UseO : UseO links the concept Use to the concept Object.
  - UseV : UseV links the concept Use to the concept View.

- **Classification definition axiom:**
  $LevelV \sqsubseteq \top$
  $Attribute \sqsubseteq AttributeV.View \sqcap AttributeL.LevelV$
  It defines relationship between the view and its classification level, where:
  AttributeV and AttributeL are binary relations such as:
  - AttributeV : AttributeV links the concept Attribute to the concept View.
  - AttributeL : AttributeL links the concept Attribute to the concept LevelV.

- **Clearance definition axiom:**
  $LevelL \sqsubseteq \top$
  $Assign \sqsubseteq AssignR.Role \sqcap AssignL.LevelR$
  It defines relationship between the view and its classification level, where:
  AssignR and AssignL are binary relations such as:
  - AssignR : AssignR links the concept Assign to the concept Role.
  - AssignL : AssignL links the concept Assign to the concept LevelR.

Depending on the model of Bell and La Padula [28], there are two types of authorization, one for reading permission and another one for writing permission. And because we have two levels in our model: abstract and concrete level, we will give axioms for abstract permission (reading and writing) and axioms for concrete permission (reading and writing).

To define a default permission, we use the following axioms.

- **Reading Permission attribution axiom:** defines the relation between role and view. A default reading permission is given to role **R** on a view **V** when its clearance level is greater than or equal to the classification level of the view.

  $\delta RPermission \sqsubseteq RPermissionR.Role \sqcap RPermissionV.View \sqcap Attribute \sqcap Assign \sqcap LevelR$

  Where:
  RPermissionR and RPermissionV are binary relations such as:
  - RPermissionR : RPermissionR links the concept RPermission to the concept Role.
  - RPermissionV : RPermissionV links the concept RPermission to the concept View.
  - LevelR $\sqsubseteq$ haslevel At-least LevelV

- **Writing Permission attribution axiom:** defines the relation between role and view. A default writing permission is given to role **R** on a view **V** when its clearance level is less than or equal to the classification level of the view.

  $\delta WPermission \sqsubseteq WPermissionR.Role \sqcap WPermissionV.View \sqcap Attribute \sqcap Assign \sqcap LevelR$

  Where:
  WPermissionR and WPermissionV are binary relations such as:
  - WPermissionR : WPermissionR links the concept WPermission to the concept Role.
  - WPermissionV : WPermissionV links the concept WPermission to the concept View.
  - LevelR $\sqsubseteq$ haslevel At-most LevelV

A concrete permission is expressed with the next axioms.

- **Concrete Reading Permission axiom:**
  $Is-Rpermitted \sqsubseteq Is-RpermittedS.Subject \sqcap Is-RpemittedO.Object$

  A concrete reading permission is given to subject **S** on an object **O**, where:
  Is-RpermittedS, Is-RpermittedO are binary relations such as:
  - Is-RpermittedS : Is-RpermittedS links the concept Is-Rpermitted to the concept Subject.
  - Is-RpermittedO : Is-RpermittedO links the concept Is-Rpermitted to the concept Object.

- **Concrete Writing Permission axiom:**
  $Is-Wpermitted \sqsubseteq Is-WpermittedS.Subject \sqcap Is-WpemittedO.Object$

  A concrete writing permission is given to subject **S** on an object **O**, where:
  Is-WpermittedS, Is-WpermittedO are binary relations such as:
  - Is-WpermittedS : Is-WpermittedS links the concept Is-Wpermitted to the concept Subject.
  - Is-WpermittedO : Is-WpermittedO links the concept Is-Wpermitted to the concept Object.

Definition of rules of security:
$\delta Is-Rpermitted \sqsubseteq Employ \sqcap Use \sqcap \delta RPermission$
$\delta Is-Wpermitted \sqsubseteq Employ \sqcap Use \sqcap \delta WPermission$

- If a subject **S** is employed in a role **R** *(Employ)*, and if there is a relation between an object **O** and a view **V** *(Use)*, and if we have a default reading permission (resp. default writing permission) relation between role **R** and a view **V** *($\delta RPermission$)* (resp. *($\delta WPermission$)*), we deduce that a subject **S** is by default permitted to perform action of reading (resp. writing) on object **O** *($\delta Is-Rpermitted$)* (resp. $\delta Is-Wpermitted$), and because $Is-Rpermitted \sqsubseteq \delta Is-Rpermitted$ (resp. $Is-Wpermitted \sqsubseteq \delta Is-Wpermitted$) (a concrete permision can be deduced from a default permission), we can finally say that a subject **S** is permitted to perform action of reading (resp. writing) on object **O**.

$Is-Rpermitted^{\epsilon} \sqsubseteq Employ \sqcap Use \sqcap RPermission^{\epsilon}$
$Is-Wpermitted^{\epsilon} \sqsubseteq Employ \sqcap Use \sqcap WPermission^{\epsilon}$

- By cons, if we have an exception on a reading permission concept wrote $RPermission^{\epsilon}$ (resp. wrinting permission concept wrote $WPermission^{\epsilon}$) , we say that we have an exception on a concept Is-Rpermitted wrote $Is-Rpermitted^{\epsilon}$ (resp. exception on a concept Is-Wpermitted wrote $Is-Wpermitted^{\epsilon}$) , and because $Is-Rpermitted \not\sqsubseteq Is-Rpermitted^{\epsilon}$ (resp. $Is-Wpermitted \not\sqsubseteq Is-Wpermitted^{\epsilon}$) (a concrete permission can not be deduced from an exceptional permission), we can deduce that a subject **S** is prohibited to perform action of reading (resp. writing) on object **O**.

B. The ABox

It contains statment about individuals. We could have many ABox for one TBox depending on applications. We illustrate this in the next section, we show how a security policy can be handled by our tool and how we can infer authorizations.

- Using instances of Table 3, the system cannot infer that **Jean** has the **default permission** to **read** the **PS1** because the classification level of the role Secretary which is played by Jean is less than the clearance level of the view Project-statistics. And because the default permission cannot be deduced, the concrete permission cannot also be deduced.

Suppose now that the assistant is absent, and the director needs statistics of the project.

The context now is different, and it is considered as an exception to the default one. We can give the secretary a temporary permission by changing his classification level, this changement is only valid in this context.

We can now deduce that **Jean** has the **default permission** to **read** the **PS1** because the context ***Absence of the Assistant*** is true.

Then we add this instance to the ABox : $\delta Permission(P1)$.

Where:

$\delta RPermission(P1) \sqsubseteq RPermisionV.View(Project-statistics) \sqcap RPermissionR.Role(Secretary) \sqcap Attribute(At2) \sqcap Assign(As4) \sqcap LevelR$

ABox
Role(Director);
Role(Assistant);
Role(Secretary);
Subject(Adam);
Subject(Sara);
Subject(Jean);
View(Project-contract);
View(Project-statistics);
View(Project-description);
Object(PC1);
Object(PS1);
Object(PD1);
LevelR(Secret);
LevelR(Confidential);
LevelR(Public);
LevelV(Secret);
LevelV(Confidential);
LevelV(Public);
$Employ(E1) \sqsubseteq EmployS.Subject(Adam) \sqcap EmployR.Role(Director)$;
$Employ(E2) \sqsubseteq EmployS.Subject(Sara) \sqcap EmployR.Role(Assistant)$;
$Employ(E3) \sqsubseteq EmployS.Subject(Jean) \sqcap EmployR.Role(Secretary)$;
$Use(U1) \sqsubseteq UseO.Object(PC1) \sqcap UseV.view(Project - contract)$;
$Use(U2) \sqsubseteq UseO.Object(PS1) \sqcap UseV.view(Project - statistical)$;
$Use(U3) \sqsubseteq UseO.Object(PD1) \sqcap UseV.view(Project - description)$;
$Attribute(At1) \sqsubseteq AttributeV.View(Project - contract) \sqcap AttributeL.LevelV(Secret)$;
$Attribute(At2) \sqsubseteq AttributeV.View(Project - statistics) \sqcap AttributeL.LevelV(Confidential)$;
$Attribute(At3) \sqsubseteq AttributeV.View(Project - description) \sqcap AttributeL.LevelV(Public)$;
$Assign(As1) \sqsubseteq AssignR.Role(Director) \sqcap AssignL.LevelR(Secret)$;
$Assign(As2) \sqsubseteq AssignR.Role(Assistant) \sqcap AssignL.LevelR(Confidential)$;
$Assign(As3) \sqsubseteq AssignR.Role(Secretary) \sqcap AssignL.LevelR(Public)$;

TABLE III
ABOX

and, $Assign(As4) \sqsubseteq AssignR.Role(Secretary) \sqcap AssignL.LevelR(Confidential)$

- **Access control if the context Absence of the Assistant is true**: Suppose that user Jean want to read the PS1, can he obtain that privilege?
  We know that:
  - Jean plays the role of Secretary: Employ(E3);
  - and, PS1 is an object used in the view Project-statistics: Use(U2);
  - and, in this context, Secretary has a clearance level equal to Confidential: Assign(As4);
  - and, Project-statistics has a classification level equal to Confidential: Attribute(At2);
  - and finally, by default, each person who plays the role of Secretary is permitted to consult Project-statistics when the assistant is absent: $\delta RPermission(P1)$.
  Formally, we write:
  $Employ(E1) \sqcap Use(U1) \sqcap \delta RPermission(P1)$
  Using security rules, we can deduce that the preceding proposition subsumes $\delta Is - Rpermitted(I1)$.
  Where:
  $Is - Rpermitted(I1) \sqsubseteq Is - RpermittedS.Subject(Sara) \sqcap Is - RpemittedO.Object(DB - Exam)$
  And because $Is - Rpermitted(I1) \sqsubseteq \delta Is - Rpermitted(I1)$, we can deduce that Jean is permitted to read PS1 if the assistant is absent.

- **suppose that the assistant is absent and a substitute was brought**: can Jean read PS1?
  In the context **Assistant absent + substitute present** , the system deduce a new instance P2 and we add to the ABox the next rule:
  $Permision(P1)^{\epsilon} \sqsubseteq \delta Permission(P2)$
  We know that:
  - Jean plays the role of Secretary: Employ(E3);
  - and, PS1 is an object used in the view Project-statistics: Use(U2);
  - and, in this context, Secretary has a clearance level equal to Confidential: Assign(As4);
  - and, Project-statistics has a classification level equal to Confidential: Attribute(At2);
  - and finally, by default, each person who plays the role of Secretary is permitted to consult Project-statistics when the assistant is absent and there is a new substitute: $\delta RPermission(P2)$.

  We obtain:
  $Employ(E3) \sqcap Use(U2) \sqcap \delta Permission(P2)$
  $\equiv Employ(E3) \sqcap Use(U2) \sqcap \delta Permission(P1)^{\epsilon}$
  We know that $A^{\epsilon} \equiv \delta A^{\epsilon}$, we obtain:
  $\equiv Employ(E3) \sqcap Use(U2) \sqcap Permission(P1)^{\epsilon}$

  Using security rules, we can deduce that the precedent

proposition subsumes $Is - permitted(I1)^{\epsilon}$.

And, because $Is - permitted(I1) \not\sqsubseteq Is - permitted(I1)^{\epsilon}$, we cannot deduce Is-permitted(I1). Therefore Jean is not permitted to read PS1 when there is a substitute to the absent assistant.

Our policy language allows us to have more than one exception in a context. Exception at an even level cancel the effects of exceptions and therefore infers the property by default [3].

Supose that we have a disjunction of context, for example "absence of assistant or absence of assistant with substitute present", here we can use the connective "lcs" to deduce permission

- **lcs(absence of assistant, absence of assistant with substitute present)**: Suppose that user Jean wants to read PS1; can he obtain that privilege?
  We know that:
  - Jean plays the role of Secretary: Employ(E3);
  - and, PS1 is an object used in the view Project-statistics: Use(U2);
  - and, in this context, Secretary has a clearance level equal to Confidential: Assign(As4);
  - and, Project-statistics has a classification level equal to Confidential: Attribute(At2);
  - and we have the two previous permissions Permission(P1) and Permission(P2), defined respectively for the context (absence of assistant) and context (absence of assistant with substitute present).

  We obtain:
  $Employ(E3) \sqcap Use(U2) \sqcap lcs(\delta Permission(P1), \delta Permission(P2))$
  $\equiv Employ(E3) \sqcap Use(U2) \sqcap lcs(\delta Permission(P1), \delta Permission(P1)^{\epsilon})$

  using lcs properties, we obtain:
  $\equiv Employ(E3) \sqcap Use(U2) \sqcap \delta Permission(P1)$
  Using security rules, we can deduce that the precedent proposition subsumes $\delta Is - permitted(I1)$.

  And, because $Is - permitted(I1) \sqsubseteq \delta Is - permitted(I1)$, we can deduce Is-permitted(I1). Therefore Jean is permitted to read SP1 when one of these contexts is true (absence of assistant, absence of assistant with substitute present).

VII. CONCLUSION AND FUTURE WORK

The work presented in this paper has led to the definition of a new system based on description logic that is expressive enough to be used as part of an application and to represent default knowledge and exceptional knowledge. The $JClassic^{+}_{\delta\epsilon}$ highlights the interests and the relevance of defaults in conceptual definition. For the $JClassic^{+}_{\delta\epsilon}$ language, we have given a set of axioms outlining the essential properties of the connectives from this definitional point of view: property links default characteristics to exceptional or strict ones. This set of axioms induces a class of $CL^{+}_{\delta\epsilon}$-algebra of which the terms are concept descriptions. Using the conjunction connectives $\sqcap$ and "lcs", the set of concept can be partially ordered w.r.t the equational system (descriptive subsumption in free algebra). $JClassic^{+}_{\delta\epsilon}$ is defined with a universel algebraic corresponding to a denotational semantic, where terms are denoted exactly by sets of strict and default properties.

This system consists of three modules: a module for representing knowledge, a module to use that knowledge and a module to update knowledge. The module which allows to use knowledge is endowed with a subsumption algorithm which is correct, complete and polynomial.

In our work, the description logic is endowed with an algebraic intensional semantics, in which concepts are denoted by a normal form of all their properties. These normal forms (i.e., elements of the intensional semantic) are used directly as an input to the algorithm of subsumption and algorithm of deductive inferences.

The developed tool has been used to describe our contextual access control model in which authorization is assigned to a subject according to its role in an organization in a given context. The two operators of default and exception are used in a clever way to assign permission depending on the context. Each time, the context changes, permissions are redefined and re-assigned to subjects.

Context can take several values, it can be a default one, an exception to the actual context, conjunction of contexts or disjunction of context. In this paper, we specially lay emphasis upon the last one.

An interseting topic for future research is to extend our tool to take into account spacial-temporal context to make our system more expressive with keeping a reasonable complexity. We also envisage to explore other appropriate and real applications.

REFERENCES

[1] N. Boustia and A. Mokhtari. $JClassic^{+}_{\delta\epsilon}$: A Description Logic Reasoning Tool: Application to Dynamic Access Control. In *Proc. The Second International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking*, Computation Tools'11, September 25-30, 2011, Rome, pp. 25-30, ISBN: 978-1-61208-159-5.

[2] D. E. Denning. Multilevel secure database systems: Requirements and model. In *NAS/AFSB Summer Study on Multilevel Database Management Security, Working Paper*, June 1982.

[3] F. Coupey and C. Fouqueré. Extending conceptual definitions with default knowledge. *Computational Intelligence*, vol 13, no 2, pp. 258-299 , 1997.

[4] http://www.cs.man.ac.uk/ sattler/reasoners.html. April, 2012.

[5] F.M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. In *Principles of language representation and reasoning: second international conference*, pp. 151-162, 1991.

[6] N. Boustia and A. Mokhtari. A Contextual Multilevel Access Control Model. In *Int. J. Internet Technology and Secured Transactions*, Vol. 3, No. 4, pp. 354-372, 2011.

[7] N. Boustia and A. Mokhtari. A dynamic access control model. In *Applied Intelligence Journal*, Volume 36, Number 1, pp. 190-207, DOI 10.1007/s10489-010-0254-z, 2012.

[8] A. Borgida and P.F. Patel-Schneider. A Semantic and Complete algorithm for subsumption in the CLASSIC description logic. *Artificial Intelligence Research*, vol 1, pp. 277-308, 1994.

[9] W.W. Cohen, A. Borgida and H. Hirsh. Computing Least Common Subsumer in Description Logic. In *10th National Conference of the American Association for Artificial Intelligence*, pp. 754-760, San Jose, California, 1992.

[10] V. Ventos and P. Brésellec. Least Common Subsumption as a connective. In *Proceeding of International Workshop on Description Logic*, Paris, France, 1997.

[11] V. Ventos, P. Brésellec and H. Soldano. Explicitly Using Default Knowledge in Concept Learning: An Extended Description Logics Plus Strict and Default Rules. In *Logic Programming and Nonmonotonic Reasoning, 6th International Conference*, LPNMR 2001, pp. 173-185, Vienna, Austria, September 17-19, 2001.

[12] A. Corradi, R. Montanari, and D. Tibaldi. Context-Based Access Control in Ubiquitous Environments. In *3rd IEEE International Symposium on Network Computing and Applications (NCA)*, August 2004.

[13] S. Fu and C.-Z Xu. A Coordinated Spatio-Temporal Access Control Model for Mobile Computing in Coalition Environments. In *In 19th IEEE International Parallel and Distributed Processing Symposium*, April 2005.

[14] F. Hansen and V. Oleshchuk. Spatial Role-Based Access Control Model for Wireless Networks. In *IEEE 58th Vehicular Technology Conference*, VTC 2003-Fall, volume 3, October 2003.

[15] M. Strembeck and G. Neumann. An Integrated Approach to Engineer and Enforce Context Constraints in RBAC Environements. In *ACM transactions on information and System Security*, 7(3), pp.392-427, 2004.

[16] H. Wedde and M. Lischka. Role-Based Access Control in Ambient and Remote Space. In *9th ACM Symposium on Access Control Models and Technologies (SACMAT 2004)*, USA, June 2004.

[17] G. Zhang and M. Parashar. Dynamic Context-aware Access Control for Grid Applications. In *4th International Workshop on Grid Computing*, November 2003.

[18] J.B.D. Joshi, E. Bertino, and A. Ghafoor Generalized Temporal Role-Based Access Control Model. 17(1), pp. 4-23, January 2005.

[19] C.K. Georgiadis, I. Mavridis, G. Pangalos and R.K. Thomas. Flexible Team-based Access Control Using Contexts. In *Sixth ACM Symposium on Access Control Models and Technologies*, pp. 21-27. ACM Press, Chantilly, 2001.

[20] J. Hu and A.C. Weaver. A Dynamic, Context-Aware Security Infrastructure for Distributed Healthcare Applications. In *First Workshop on Pervasive Privacy Security, Privacy, and Trust*, Boston, MA, USA, 2004, http://www.pspt.org/techprog.html

[21] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel, and G. Trouessin. Organization Based Access Control. In *4th IEEE International Workshop on Policies for Distributed Systems and Networks (Policy'03)*, Lake Come, Italie, June 2003.

[22] F. Cuppens and A. Miège. Modelling Contexts in the ORBAC Model. In *19th Annual Computer Security Applications Conference*, Las Vegas, December 2003.

[23] H. Debar, Y. Thomas, F. Cuppens and N. Cuppens-Boulahia. Enabling Automated Threat Response through the Use of a Dynamic Security Policy In*"Journal in Computer Virology (JCV)*, 3, 3 (2007), pp. 195-210, 2007.

[24] N. Boustia and A. Mokhtari. Representation and reasoning on ORBAC: Description Logic with Defaults and Exceptions Approach. In *Workshop on Privacy and Security - Artificial Intelligence (PSAI)*, pp. 1008-1012, ARES'08, Spain, 2008.

[25] N. Boustia and A. Mokhtari. $DL_{\delta\epsilon} - OrBAC$: Context based Access Control. In *Proc. WOSIS'09*, pp. 111-118, Italy, 2009.

[26] R.J. Brachman, D.L. McGuinness, P.F. Patel-Schneider, L. Alperin Resnick, and A. Borgida. Living with CLASSIC: When and How to Use a KL-ONE-Like Language. In *John Sowa, ed., Principles of Semantic Networks: Explorations in the representation of knowledge*, pp. 401-456, Morgan-Kaufmann: San Mateo, California, 1991.

[27] R.J. Brachman, D.L. McGuinness, L. Alperin Resnick, and A. Borgida. CLASSIC: A Structural Data Model for Objects. In *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*, pp. 59-67, June 1989.

[28] D.E. Bell, and L.J. LaPadula Secure Computer System: Unified Exposition and Multics Interpretation. In *MITRE Technical Report MTR-2997*, July, 1975.