UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

Tesi di Laurea

ESTENSIONE DEL LINGUAGGIO FACPL PER ESPRIMERE POLITICHE DI GESTIONE DELL'UTILIZZO CONTINUATIVO DELLE RISORSE DI UN SISTEMA DI CALCOLO

EXTENSION OF LANGUAGE FACPL TO USE ACCESS CONTROL POLICIES BASED ON CONTINUATIVE USE OF RESOUCES

FILIPPO MAMELI

Relatore: *Rosario Pugliese*
Correlatore: *Andrea Margheri*

Anno Accademico 2015-2016

Filippo Mameli: *Estensione del linguaggio FACPL per esprimere politiche di gestione dell'utilizzo continuativo delle risorse di un sistema di calcolo,* Corso di Laurea in Informatica, © Anno Accademico 2015-2016

INDICE

1

INTRODUZIONE

Dalla loro nascita i sistemi informatici hanno avuto il ruolo di gestore di dati. Il tipo di queste informazioni ha reso necessario l' utilizzo di un sistema che le proteggesse. I dati più sensibili se diffusi senza una valida autorizzazione possono arrecare danni economici ad una società o anche nuocere gli utenti nel privato. Lo sviluppo del web ha generato un interconnessione ancora più forte tra i sistemi e questo ha messo ancora più a rischio le informazioni più critiche.

1.1 ESTENSIONE DEL LINGUAGGIO

2

ACCESS CONTROL E USAGE CONTROL

2.1 CONTROLLO DEGLI ACCESSI

La protezione dei dati ha determinato la necessità di creare strumenti per il controllo degli accessi che potevano eliminare ,o almeno limitare, i rischi derivati dalla perdita delle informazioni.
Nel corso del tempo si sono sviluppati alcuni modelli per i sistemi del controllo degli accessi. A seconda delle necessità sono stati adottati numerosi tipi di tecnologie[1]. Nelle sezioni successive se ne presentano alcune.

2.1.1 *Access Control List*

Access Control List(ACL) è stato creato agli inizi degli anni settanta per la necessità di un controllo degli accessi sui sistemi multiutente. Utilizza una lista di utenti con annesse le possibili azioni autorizzate. Il modello è molto semplice, ma ha numerose limitazioni. Quando nel sistema ci sono numerosi utenti o risorse, la quantità di dati da verificare diventa difficile da gestire. Questo può portare a errori di assegnazione di autorizzazioni e ad un eccesivo numero di controllo necessari per un singolo accesso.

2.1.2 *Role Based Access Control*

Role Based Access Control (RBAC) è l' evoluzione di ACL. In questo modello vengono introdotti i *ruoli*. Più utenti possono avere lo stesso ruolo e quindi avere a disposizione le tutte risorse connesse a questo. Il modello diventa scalabile e più facile da gestire, inoltre si possono anche creare delle gerarchie per facilitare l' assegnamento di risorse in base alla classificazione dell' utente.

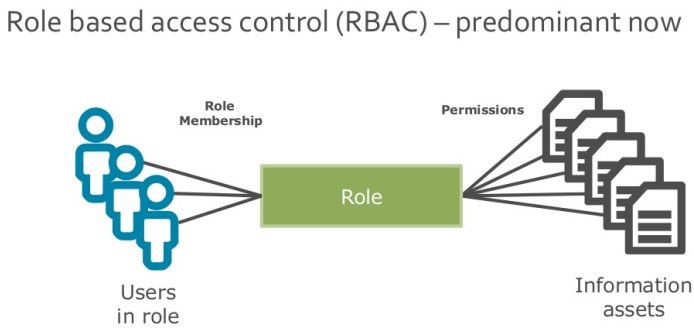

Figura 1: RBAC

2.1.3 *Attribute Based Access Control*

Attribute Based Access Control (ABAC) si basa sull'utilizzo di attributi associati all'utente, all'azione o al contesto della richiesta. La valutazione di una autorizzazione diventa più specifica e le regole sono più precise per ogni risorsa. Questo tipo di modello non è utilizzato nei sistemi operativi,dove ACL e RBAC sono i modelli più diffusi, ma è sviluppatto spesso a livello applicativo. Il problema fondamentale di questo paradigma è che le regole non sono uniformi e se il numero di risorse è consistente, la gestione di queste diventa complicata. Il modello Policy Based Access Control cerca di risolvere il difetto di ABAC.

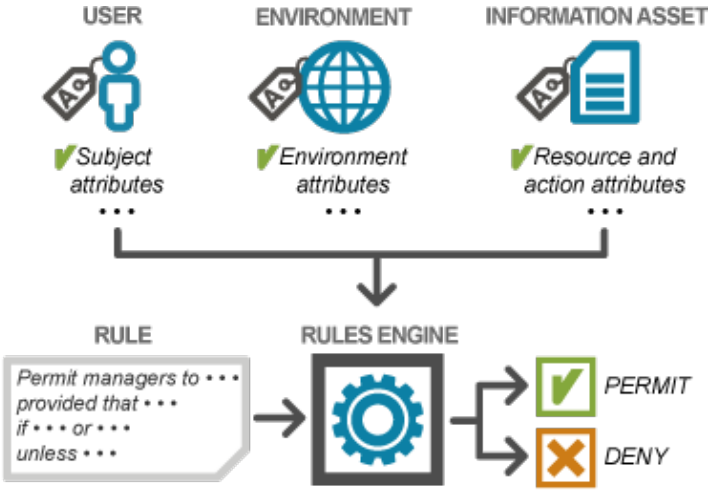

Figura 2: ABAC

2.1.4 *Policy Based Access Control*

Policy Based Access Control riorganizza il modello ABAC per semplificare la gestione delle regole. Il sistema si basa su *politiche* che non sono altro che insiemi di *regole*. A ogni regola è associato un attributo che l'utente deve avere e ogni politica valuta tutte le regole nel suo insieme per creare la risposta sull'autorizzazione. Anche le politiche possono essere messe insieme per creare gruppi di politiche, in questo modo il sistema diventa scalabile e di più facile utilizzo.
Per costruire un sistema di controllo degli accessi basato sul modello PBAC è necessario l'utilizzo di un linguaggio adatto allo scopo. L'organizzazione OASIS (Organization for the Advancement of Structured Information Standards) ha creato il linguaggio eXtensible Access Control Markup Language (XACML) che è diventato lo standard per lo sviluppo di un sistema costruito sul modello PBAC.

2.2 USAGE CONTROL

Dopo quaranta anni di studi sul controllo degli accessi i modelli sviluppati si sono consolidati e sono largamente utilizzati su sistemi operativi o applicazioni. Tuttavia la complessità e la varietà degli ambienti informatici moderni va oltre i limiti dei modelli creati.
Il termine Usage Control (UCON) è stato ripreso da Jaehong Park e Ravi Sandhu per creare il modello $UCON_{ABC}$[2], questo è una generalizzazione dell'Access Control che include obbligazioni, condizioni sull'utilizzo, controlli continuativi e mutabilità. Comprende e migliora i modelli di controllo di accesso tradizionali, quali Trust Management (TM) e Digital Rights Management (DRM) aggiungendo la gestione di attributi variabili e la continuità nella valutazione delle decisioni per l'accesso. Il modello $UCON_{ABC}$ estende i controlli sull'accesso tradizionali ed è composto da otto componenti fondamentali. Queste sono subjects,subject attributes, objects,objects attributes, rights, authorizations, obligations e conditions.
I Subjects sono entità a cui si associano degli attributi e hanno o esercitano Rights sugli Objects. Possiamo per semplicità associare i Subjects ad un singolo individuo umano.
Gli Objects sono insiemi di entità su cui i Subjects possono avere dei Rights, questi possono essere usati o vi si può fare accesso. Possono essere associati ad esempio a un libro , o a una qualsiasi risorsa.
I Rights sono i privilegi che i Subjects hanno o esercitano sugli Objects.
I tre fattori Authorizations, oBligations e Conditions(da cui prende anche

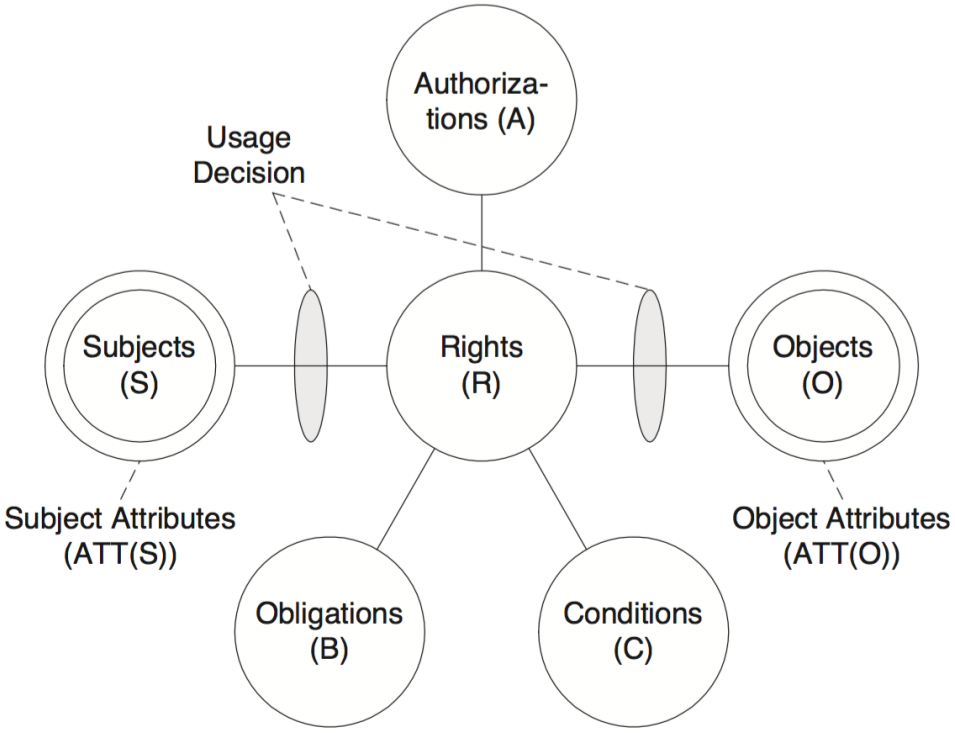

Figura 3: UCON

il nome il modello) sono predicati funzionali che devono essere valutati per le decisioni sull'uso. I tradizionali Access Controls utilizzano solo le Authorizations per il processo di decisione, Obligations e Conditions sono i nuovi componenti che entrano a far parte della valutazione.
Le Authorizations devono valutare la decisione sull'uso. Queste danno un responso positivo o negativo a seconda che la domanda di un Subject sia accettata o meno.
Le Obligation verificano i requisiti obbligatori che un Subject deve eseguire prima o durante l'utilizzo di una risorsa.
Infine le Condition restituiscono true o false in base alle variabili dell'ambiente o allo stato del sistema.
Il processo di decisione è diviso in tre fasi[3]:Before usage(pre), Ongoing usage(on) e After usage. La valutazione della prima parte inizia da una richiesta e non ha differenze con il processo valutativo dell'Accesso Control. Nella seconda invece si utilizzano i nuovi predicati introdotti ed è in questa parte che si affermano i controlli continuativi, le obbligazioni e le condizioni sull'utilizzo.

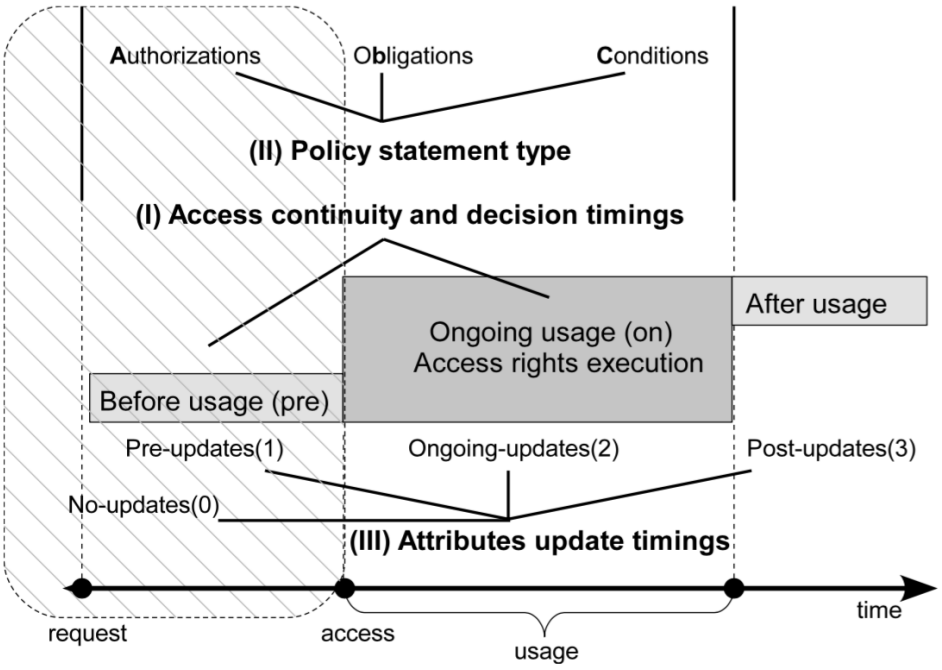

Figura 4: Fasi del processo di decisione

L'ultima parte varia in base agli eventi delle fasi precedenti. Ad esempio se il Subject che ha richiesto un accesso ha violato una policy oltre al non aver ricevuto l'autorizzazione potrebbe anche essere ammonito e il sistema potrebbe non accettare più nessuna sua richiesta.

Esempi di Usage Control

3

LINGUAGGIO FACPL

3.1 LINGUAGGIO FACPL

FACPL

3.1.1 *Sintassi*

3.1.2 *Componenti del sistema*

Target

Obligation

PDP

PEP

3.2 ESEMPIO

Tabella 1: Sintassi di FACPL

Policy Authorisation Systems	PAS ::=	(pep : $EnfAlg$ pdp : PDP)
Enforcement algorithms	$EnfAlg$ ::=	base | deny-biased | permit-biased
Policy Decision Points	PDP ::=	$\{Alg$ policies : $Policy^+\}$
Combining algorithms	Alg ::=	p-over$_\delta$ | d-over$_\delta$ | d-unless-p$_\delta$ | p-unless-d$_\delta$ | first-app$_\delta$ | one-app$_\delta$ | weak-con$_\delta$ | strong-con$_\delta$
fulfilment strategies	δ ::=	greedy | all
Policies	$Policy$ ::=	$(Effect$ target : Expr obl : $Obligation^*$) | $\{Alg$ target : Expr policies : $Policy^+$ obl : $Obligation^*\}$
Effects	$Effect$ ::=	permit | deny
Obligations	$Obligation$ ::=	$[Effect\ ObType$ PepAction(Expr*)$]$
Obligation Types	$ObType$ ::=	M | O
Expressions	$Expr$ ::=	$Name$ | $Value$ | and($Expr, Expr$) | or($Expr, Expr$) | not($Expr$) | equal($Expr, Expr$) | in($Expr, Expr$) | greater-than($Expr, Expr$) | add($Expr, Expr$) | subtract($Expr, Expr$) | divide($Expr, Expr$) | multiply($Expr, Expr$)
Attribute Names	$Name$ ::=	$Identifier/Identifier$
Literal Values	$Value$ ::=	true | false | $Double$ | $String$ | $Date$
Requests	$Request$ ::=	$(Name, Value)^+$

Tabella 2: Sintassi ausiliaria per le risposte

PDP Responses	$PDPResponse$ ::=	$\langle Decision\ \ FObligation^*\rangle$
Decisions	$Decision$ ::=	permit | deny | not-app | indet
Fulfilled obligations	$FObligation$ ::=	$[ObType$ PepAction($Value^*$)$]$

4

IMPLEMENTAZIONE USAGE CONTROL IN FACPL

5

ESEMPI

5.1 CONTATORE

5.2 DATA

5.3 LETTURA E SCRITTURA

6

STRUMENTI USATI PER LO SVILUPPO

6.1 xtext

6.2 plugin eclipse

7

CONCLUSIONI

7.1 SVILUPPI FUTURI

BIBLIOGRAFIA

[1] NIST - *A survey of access Control Models* - http://csrc.nist.gov/news_events/privilege-management-workshop/PvM-Model-Survey-Aug26-2009.pdf (Cited on page 5.)

[2] Jaehong Park, Ravi Sandhu - *The UCON Usage Control Model* - http://drjae.com/Publications_files/ucon-abc.pdf (Cited on page 7.)

[3] Aliaksandr Lazouski, Fabio Martinelli, Paolo Mori - *Usage control in computer security: A Survey* (Cited on page 8.)

[4] Aliaksandr Lazouski, Gaetano Mancini, Fabio Martinelli, Paolo Mori - *Usage Control in Cloud Systems* - Istituto di informatica e Telematica, Consiglio Nazionale delle Ricerche.

[5] Alexander Pretschner, Manuel Hilty, Florian Schutz, Christian Schaefer, Thomas Wlater - *Usage Control Engorcement*

[6] Andrea Margheri, Massimiliano Masi, Rosario Pugliese, Francesco Tiezzi - *A Formal Framework for Specification, Analysis and Enforcement of Access Control Policies*

[7] Jaehong Park, Ravi Sandhu - *A Position Paper: A Usage Control (UCON) Model for Social Networks Privacy*

[8] Leanid Krautsevich, Aliaksandr Lazouski, Fabio Martinelli, Paolo Mori, Artsiom Yautsiukhin - *Usage Control, Risk and Trust*