A.A. 2006-2007 Corso di
Sicurezza dei sistemi informatici 1

Note integrative alla Lezione 15

Università di Catania
Corso di Laurea in Informatica applicata
(Comiso)

10/05/2007

Giuseppe Scollo

2. Modelli di autorizzazione negli ambienti di collaborazione

It is easier to get forgiveness than permission. [1]
Stewart's Law of Retroaction

La problematica dell'autorizzazione nei sistemi di supporto alla collaborazione riscuote una crescente attenzione da parte dei progettisti di tali sistemi. Ancora solo alcuni anni fa, lo stato dell'arte induceva [5] a denunciare la sostanziale assenza di adozione, in sistemi per il *Computer-Supported Collaborative Work* (CSCW) di largo uso, di alcuno dei modelli di controllo dell'accesso specificamente pensati per ambienti di collaborazione, già proposti allora nella letteratura scientifica.

Oggi la situazione è radicalmente diversa: nessun sistema CSCW può competere in efficacia con quelli di maggiore diffusione senza offrire funzionalità di supporto ad autorizzazione e controllo di accesso che siano comparabili a quelle presenti in tali sistemi. Una ragione per un così rapido progresso è certamente rintracciabile nella crescente importanza delle tematiche legate alla sicurezza nei sistemi distribuiti. Una ragione più specifica risiede nel fatto che si è capito che l'inadeguatezza delle suddette funzionalità in un sistema CSCW di fatto ne cambia la natura, attraverso una mutazione dell'acronimo in cui la "S" si fa "O": da *Supporto* ad *Ostacolo* alla collaborazione.

Ma cosa rende *adeguato* un sistema di autorizzazione e controllo di accesso in un ambiente collaborativo? Scopo precipuo di questo capitolo è, appunto, dare risposta a questa domanda. A tal fine, il resto del capitolo è organizzato come segue.

- Si distingue preliminarmente, in sez. 2.1, fra *autorizzazione* e *controllo di accesso*, in quanto diverse sono le rispettive caratteristiche dei due concetti che contribuiscono alla risposta che ci proponiamo di determinare. Le caratteristiche salienti del secondo vengono enunciate subito, mentre il resto del capitolo tratta dei modelli di autorizzazione che il controllo di accesso rende operativi.

[1] È più facile ottenere perdono che permesso.

- I concetti fondamentali che un adeguato modello di autorizzazione deve rendere disponibili in un sistema di supporto alla collaborazione sono presentati in sez. 2.2. Si tratta in tutti i casi di astrazioni nel più intuitivo senso matematico del termine, cioè di *astrazioni insiemistiche*, che formalizzano con notevole semplicità concetti connaturati all'autorizzazione, quali il *raggruppare* le operazioni eseguibili su vari tipi di oggetti in insiemi classificati da *tipi di diritti di accesso* e il definire *ruoli* in termini di questi ultimi.
- La naturale estensione dei meccanismi di astrazione insiemistica suddetti per gli ambienti collaborativi consiste nel raccoglierne gli utenti in insiemi detti *gruppi*, associati a spazi di lavoro e dotati di diritti di accesso in essi attraverso l'assegnazione di ruoli. Il modello di autorizzazione, basato su questi concetti, in uso nel sistema *BSCW* [1] è spiegato in sez. 2.3.
- Come vedremo, è sovente possibile che un utente abbia accesso ad un oggetto o spazio collaborativo con più ruoli e/o grazie alla sua appartenenza a più gruppi. Questa molteplicità rende utile il poter *escludere* alcuni utenti o gruppi dal godimento di certi diritti di accesso a dati oggetti o spazi, indipendentemente dal conferimento ad essi di tali diritti per altra via. Si parla in tal caso di *diritti di accesso negativi*, concetto trattato in sez. 2.4.
- Un modello di autorizzazione basato sui concetti fondamentali a cui si è fatto cenno acquista semplicità dal trattare gli insiemi in gioco (diritti di accesso, ruoli, gruppi) come oggetti essi stessi, dunque accessibili e manipolabili attraverso opportune operazioni. Ciò naturalmente genera una gerarchia, potenzialmente infinita, di diritti: diritti sui diritti, diritti sui diritti sui diritti, etc. In sez. 2.5, dopo aver esaminato un semplice modo per arrestare la catena là dove è utile, cioè al secondo livello, si vedrà come questo corrisponda ad un fenomeno naturale negli ambienti collaborativi, ovvero la *delega* ad altri di determinati diritti (e responsabilità).

2.1 Autorizzazione e controllo di accesso

È bene innanzitutto distinguere fra *autenticazione* e *autorizzazione*: la prima è accertamento di veridicità dell'*identità* di qualcuno, la seconda è conferimento a qualcuno del *diritto di fare* qualcosa. Qui ci occupiamo di autorizzazione.

In secondo luogo, in una visione degli ambienti collaborativi come strutture di *oggetti* a cui utenti hanno *accesso* attraverso *operazioni*, "fare" qualcosa significa effettuare un accesso ad un oggetto attraverso una delle operazioni definite per esso. Autorizzazione è allora sinonimo di *conferimento di diritti di accesso*, distinta da (sebbene ovviamente correlata a) *controllo di accesso*, che invece consiste nel preventivo accertamento che ciascuna richiesta di accesso goda della rispettiva autorizzazione. Il controllo di accesso dunque rende *operativa* l'autorizzazione, definita ad ogni dato momento, con l'impedire che possano aver luogo accessi il cui diritto non è conferito. Nel resto di questo capitolo ci occuperemo di autorizzazione nel senso sudddetto, e più precisamente di *modelli di autorizzazione*, intesi come sistemi di concetti relativi

al conferimento di diritti di accesso. È bene però aver chiaro quali caratteristiche del sistema di controllo di accesso, che rende operativo un modello di autorizzazione, possono risultare critiche rispetto alla sua adeguatezza a tipiche esigenze presenti nei sistemi di supporto alla collaborazione.

Il controllo di accesso, per definizione, si manifesta attraverso l'esecuzione di opportuni accertamenti a fronte di ogni richiesta di accesso. È dunque inevitabile che il tempo necessario ad eseguire il controllo di accesso contribuisca al *tempo di risposta* del sistema. Questo si traduce in tempo di attesa dell'utente, dunque è evidentemente uno dei fattori di qualità da tenere in considerazione nella realizzazione di algoritmi di controllo di accesso, e nella progettazione di modelli di autorizzazione per quanto questi possano influire sulla complessità di tali algoritmi.

2.2 Tipi di diritti di accesso, ruoli

Nella sua forma più elementare, un'autorizzazione di accesso coinvolge tre entità: il *soggetto* a cui viene conferita, l'*oggetto* a cui è riferito l'accesso, l'*operazione* di accesso del soggetto all'oggetto. Una classica rappresentazione delle autorizzazioni definite in un dato stato del sistema è la *matrice di Lampson* [2], in cui si dispongono i soggetti in una dimensione e gli oggetti nell'altra: in ciascun elemento della matrice si elencano le operazioni su un dato oggetto che un dato soggetto è autorizzato a compiere in quello stato del sistema. Tradizionalmente possono seguirsi due vie per decomporre la matrice di Lampson in strutture più ridotte:

- fissare il soggetto, per ottenere la *lista delle capacità* (ingl. *capabilities*) di un dato soggetto su qualsiasi oggetto;
- fissare l'oggetto, per ottenere la *lista di controllo di accesso* (ingl. *ACL: Access Control List*) all'oggetto stesso da parte di qualsiasi soggetto.

Per le ragioni di efficienza a cui si è accennato nella precedente sez. 2.1, le rappresentazioni basate su ACL sono più diffuse (tali sono in particolare quelle dei sistemi *BSCW* [1] e *Zope* [6]): ad ogni oggetto è associata una lista che dice chi ha diritto di effettuare quali operazioni sull'oggetto.

La rappresentazione elementare, basata su ACL, dell'autorizzazione è conveniente per il controllo di accesso, ma può rapidamente diventare scomoda da usare, pur nella sua semplicità, quando si voglia permettere che (alcuni o tutti gli) utenti abbiano modo di *definire* la ACL di un oggetto. Tale opportunità si presenta spesso con la condivisione di oggetti in spazi di lavoro comune, dove di solito è disponibile al creatore dell'oggetto, ed eventualmente ad altri utenti (ad esempio, amministratori dello spazio di lavoro). La scomodità è dovuta al fatto che su un oggetto complesso possono essere definite numerose operazioni, e che il numero degli utenti che hanno accesso allo spazio comune può essere elevato. Entrambi questi fattori vengono ulteriormente potenziati quando si vogliano replicare uniformemente le ACL per un insieme

di oggetti, non necessariamente dello stesso tipo (ragion per cui la soluzione di "copiare" la ACL da un oggetto ad un altro non è sempre utilizzabile).

I seguenti concetti semplificano la definizione e gestione delle ACL. Premesso che intendiamo, in prima approssimazione, per *classe* di oggetti un insieme di oggetti sui quali sono definite le stesse operazioni,[2] definiamo:

- *tipo di diritti di accesso*: un nome, ovvero una *designazione univoca*, di un sottoinsieme delle operazioni definite per una *classe* di oggetti;
- *ruolo*: un nome, ovvero una *designazione univoca*, di un sottoinsieme delle operazioni definite su un *oggetto*.

Le due definizioni sono molto simili, ma non del tutto identiche; alla "piccola" differenza fra esse corrisponde una notevole differenza del significato intuitivo e degli usi pratici di tali concetti. Esaminiamo questa differenza: essa consiste nel fatto che il requisito di univocità della designazione

1. per i tipi di diritti di accesso, è *locale a ciascuna classe* di oggetti, dunque uno stesso tipo di diritti di accesso può designare insiemi diversi di operazioni su oggetti diversi *solo se questi sono di classi diverse*; mentre
2. per i ruoli, è *locale a ciascun oggetto*: uno stesso ruolo può designare insiemi diversi di operazioni su oggetti diversi *anche se della stessa classe*.

Per carpire il significato intuitivo dei concetti definiti sopra, e la motivazione pratica della differenza fra le rispettive definizioni, un paio di esempi dovrebbero risultare illuminanti.

Nel BSCW, sulla classe di oggetti Nota (in forum di discussione) sono definite le operazioni (elencate nella barra orizzontale in alto) confermare, inviare, copiare, ritagliare, rimuovere, archiviare; inoltre, altre operazioni (oltre a queste) sono elencate nel menù a tendina accessibile a destra del titolo di ciascuna nota: consideriamo fra queste, per semplicità, solo le operazioni modificare e rispondere. Possiamo classificare queste otto operazioni nei seguenti quattro tipi di diritti di accesso (denominati *gruppi di azioni* nel BSCW, dove però non si distingue fra i tipi *modifica* e *aggiunta*, mentre il tipo *cancellazione* è denominato *modifica estesa*):

- *lettura*: {inviare, copiare, archiviare},
- *modifica*: {confermare, modificare},
- *cancellazione*: {ritagliare, rimuovere},
- *aggiunta*: {rispondere}.

Questa classificazione si applica a tutti gli oggetti della classe Nota, cioè a tutte le note in forum di discussione. Ciascun tipo di diritto di accesso

[2] questa formulazione è una semplificazione del concetto di classe quale è inteso nella programmazione orientata agli oggetti, che risulta tuttavia consistente con tale concetto, e sufficiente nel contesto del presente discorso.

può designare un diverso insieme di operazioni per un'altra classe di oggetti, ad esempio sugli oggetti di classe Cartella è definita l'operazione nuovo_documento (per il caricamento di un documento in una cartella), che è classificabile con il tipo di diritto di accesso *aggiunta*.

Nel BSCW alcuni ruoli sono predefiniti, fra cui *Proprietario*, *Membro*, *Membro ristretto*, *Amministratore*. Sembrerebbe naturale dare al termine "ruolo" il significato di "gruppo di utenti", come ad esempio in [5], ma la definizione da noi proposta fa invece riferimento alle operazioni su oggetti. Questo si spiega con il fatto che, mentre è ben possibile assegnare ruoli ad utenti per ciascun oggetto, e dunque a ciascun ruolo per ciascun oggetto è in tal modo associato un gruppo di utenti, il significato dell'assegnazione consiste nell'attribuzione di diritti di accesso sull'oggetto agli utenti in questione. Sembra dunque più consono alla realtà identificare il concetto di ruolo con quello di designazione di un insieme di diritti di accesso, che infatti come tale risulta, per ciascun oggetto:

- definibile,
- modificabile,
- assegnabile ad utenti,

attraverso rispettive operazioni sull'oggetto.

Il fatto che i ruoli siano definibili e modificabili per ciascun oggetto spiega la differenza rispetto alla definizione di tipo di diritti di accesso: per oggetti diversi, anche se della stessa classe, uno stesso ruolo può essere diversamente definito.

Entrambi i concetti entrano in gioco nell'operazione di definizione di un (nuovo) ruolo su un oggetto nel BSCW: l'utente ha l'opportunità di immettere il *nome* del ruolo, selezionare un ruolo preesistente quale *modello*, e infine selezionare i gruppi di azioni, cioè i *tipi di diritti di accesso* secondo la nostra terminologia, da aggiungere al nuovo ruolo oltre ai diritti di accesso conferiti attraverso il ruolo modello.

Sembra lecito chiedersi quale sia il vantaggio di una tale "stratificazione su due livelli" dell'insieme di operazioni associate al nuovo ruolo: a un primo livello, raggruppamento in tipi di diritti di accesso; ad un secondo livello, selezione di un insieme di questi ultimi per il ruolo. Un primo vantaggio è evidente: la definizione di un nuovo ruolo può essere effettuata più rapidamente, grazie al ridotto numero di scelte possibili (fra *tipi* invece che fra operazioni).

Un secondo vantaggio, ancor più rilevante, della stratificazione suddetta è la possibilità di uniformità nella definizione di ruoli su oggetti di classi diverse: ad esempio, sebbene al tipo di diritti di accesso *aggiunta* corrispondano diversi insiemi di operazioni nelle classi Nota e Cartella, si potrà uniformemente selezionare tale tipo nella definizione di uno stesso ruolo (cioè di ruoli omonimi) per oggetti delle due classi: la diversità delle operazioni fra le due classi viene per così dire "mascherata" dall'omonimia dei tipi di diritti di accesso in cui sono classificate.

Un terzo vantaggio, che discende dal secondo, è di grande aiuto nella definizione automatica di ruoli e ACL per gli oggetti di nuova creazione: se non esplicitamente definiti dall'utente che crea l'oggetto, questo *eredita* ruoli e ACL dall'oggetto contenitore (cartella, forum, o altro) al cui interno viene creato; ciò è evidentemente reso possibile dall'uniformità introdotta dalla classificazione delle operazioni in tipi di diritti di accesso, che possono venire ereditati fra oggetti di classi diverse.

2.3 Modelli di autorizzazione basati sul raggruppamento

I concetti proposti nella precedente sez. 2.2 fanno prevalentemente riferimento a due delle tre entità coinvolte nell'autorizzazione: oggetto e operazione. Si è sottinteso il *soggetto*, che nel controllo di accesso è sempre e comunque un utente, cioè un individuo. Questa identificazione automatica di soggetto e individuo non è però necessariamente vera in sede di *conferimento* dell'autorizzazione su un oggetto, ad esempio al momento della sua creazione. Risulta utile a tal scopo, e del tutto connaturato agli ambienti collaborativi, considerare un *insieme di utenti* quale soggetto al quale attribuire autorizzazioni, sia direttamente, cioè in termini di diritti di accesso, sia indirettamente, cioè attraverso l'assegnazione di ruoli. Un vantaggio di questa ulteriore astrazione consiste evidentemente nel fatto che non occorre replicare l'attribuzione dell'autorizzazione per ciascun utente nell'insieme, bensì la si effettua una volta sola per tutti.

Negli ambienti collaborativi, un insieme di utenti naturalmente qualificabile come soggetto di un'autorizzazione è l'insieme degli utenti che hanno accesso ad uno spazio condiviso. Ad essi potranno attribuirsi autorizzazioni diverse attraverso il conferimento di ruoli diversi, tuttavia almeno le operazioni di lettura dovranno essere autorizzate per tutti loro, in quanto altrimenti perderebbe significato il concetto stesso di accesso allo spazio. Per ogni spazio X si definisce dunque come *gruppo degli utenti di* X l'insieme degli utenti che hanno accesso (almeno in lettura) a X.

In prima approssimazione, possiamo pensare ad una struttura di spazi condivisi come analoga a quella di un *filesystem*, organizzata in cartelle e sottocartelle. Questa è certamente la *vista* che *ciascun utente* ha di una struttura di spazi condivisi, e tale corrispondenza intuitiva ad una struttura con cui si ha già familiarità sicuramente aiuta a muovere i primi passi in un ambiente collaborativo. Tuttavia, al fine di *organizzare* e successivamente *gestire* con efficacia un sistema di spazi condivisi, in particolare ai fini della didattica in rete (ma certo non solo a questi), risulta necessario cogliere non solo l'analogia ma anche, e soprattutto, le *differenze* tra sistema di spazi condivisi e *filesystem* tradizionale. Esamineremo tali differenze innanzitutto con riferimento al caso, più semplice, del sistema *Zope*, quindi a quello più complesso del sistema *BSCW* (nonostante l'apparenza di maggiore semplicità di quest'ultimo: mai fidarsi delle apparenze!).

La struttura astratta di un *filesystem* è quella gerarchica di un *albero*, con una cartella *radice*, (sotto)cartelle ai *nodi intermedi*, e files a *nodi terminali* (o *foglie*) dell'albero.

Anche la struttura astratta di un sistema di spazi condivisi in *Zope* è un albero, con uno spazio radice e sottospazi ai nodi intermedi, dov'è dunque la differenza? Nel fatto che

1. a ciascuno spazio è associato un insieme di utenti, e
2. ciascun sottospazio *eredita* l'insieme di utenti dallo spazio che lo contiene, e *può estenderlo* con nuovi utenti.

Ne consegue che ciascun utente vede il sistema di spazi come una struttura ad albero, ma non tutti gli utenti vedono lo stesso albero: se nel sottospazio Y dello spazio X l'insieme di utenti viene esteso con nuovi utenti, questi ultimi vedono lo spazio Y come spazio radice, mentre per gli altri utenti di Y questo è un sottospazio di X.

Ad esempio, in fig. 2.1 è riconoscibile la struttura degli spazi condivisi *Zope* per uncorso a distanza via Web: nello spazio radice, i cui utenti sono gli amministratori del sistema, è definito il sottospazio tmfr, visibile anche ai docenti e tutori del corso, e in tmfr sono definiti i due sottospazi argentina e italia, rispettivamente visibili ai corsisti delle due edizioni del corso (oltre che ad amministratori, docenti e tutori).

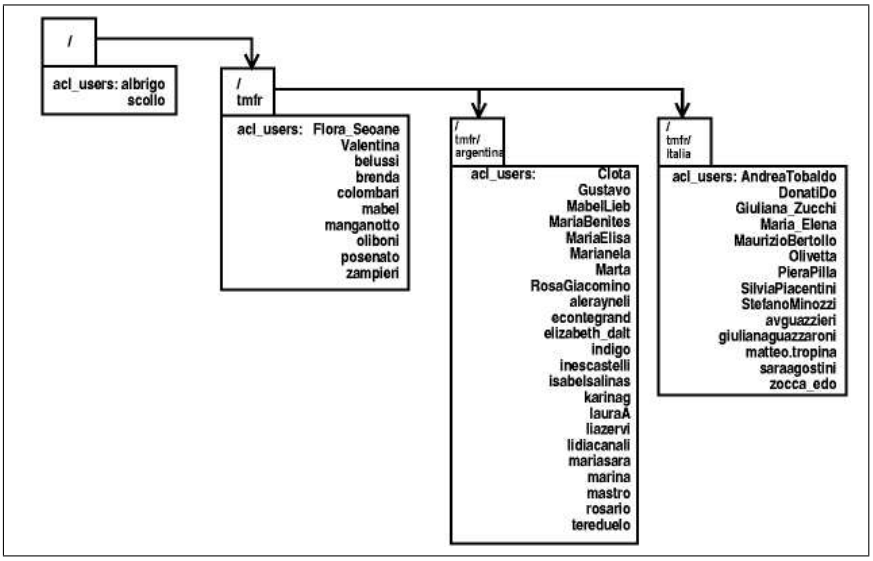

Figura 2.1. Struttura ad albero degli spazi *Zope* per un corso in rete

Una conseguenza di quanto sopra, che risulta utile tenere a mente a fini pratici e che vale anche per la più complessa struttura di spazi *BSCW*, è che

discendendo i cammini della struttura la visibilità non può diminuire.

La struttura astratta di un sistema di spazi condivisi *BSCW* si presenta come una generalizzazione di quella ad albero: è un *grafo diretto aciclico*, o *DAG* (dall'inglese *Directed Acyclic Graph*), come illustrato in fig. 2.2, cioè un diagramma costituito da un insieme di *nodi*, dove alcune coppie di nodi sono connesse da un *arco* orientato, e tale che nessun percorso lungo archi adiacenti, a partire da un qualsiasi nodo e rispettando la direzione degli archi, conduca al nodo di partenza.

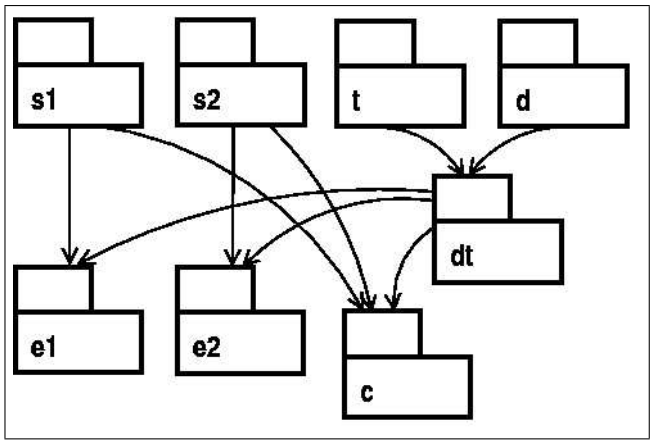

Figura 2.2. Un esempio di struttura DAG di spazi *BSCW*

Ogni albero è un DAG; in particolare, un DAG è un albero se e solo se verifica la seguente condizione: per ogni nodo tranne uno (detto nodo *radice*) esiste un solo nodo (*il* suo *predecessore*) connesso ad esso da un arco diretto verso di esso. Naturalmente, esistono DAG che non sono alberi, e ciò, per la condizione precedente, può accadere per almeno una di due ragioni (o per entrambe): 1) esiste più di un nodo privo di predecessore, 2) esiste qualche nodo con più di un predecessore. Entrambi i fenomeni si verificano comunemente nei DAG dei sistemi di spazi *BSCW*, a causa del significato di tali strutture che precisiamo come segue.

Ad ogni spazio *BSCW* è associato un *gruppo dei membri*, e ad ogni gruppo di membri *BSCW* è associato uno spazio: possiamo dunque indifferentemente interpretare i nodi del DAG come spazi o come gruppi di membri. Dal momento che vogliamo illustrare quest'ultimo concetto, scegliamo la seconda interpretazione. La differenza fondamentale tra il concetto di insieme di utenti di uno spazio *Zope* e il concetto di gruppo di membri di uno spazio *BSCW* è che membri di quest'ultimo possono essere gruppi essi stessi (però purché nessun gruppo sia membro di se stesso). Possiamo allora comprendere la struttura DAG di tali gruppi in un sistema *BSCW* come costruita a par-

tire da *gruppi individuali* {u}, {v}, ..., ciascuno dei quali consta di un solo utente: tali gruppi sono i nodi privi di predecessore nel DAG, ed a ciascuno di essi corrisponde lo spazio personale nel *BSCW*. La costruzione di gruppi più ampi è ottenuta attraverso l'operazione di condivisione di uno spazio, ovvero la sua apertura ad altri gruppi, che possono essere individuali o meno: questi diventano così membri del gruppo associato allo spazio condiviso.

Una applicazione pratica di questa tecnica di costruzione di spazi condivisi alla didattica in rete, ed i vantaggi che ne derivano, sono illustrati dal seguente problema. Un docente tiene un corso in rete per 60 alunni. La numerosità di questi obbliga la divisione del corso in due classi, ciascuna seguita da un tutore. I requisiti sulle modalità di valutazione *in itinere* prevedono che ogni settimana ciascun alunno debba svolgere individualmente (dunque non in modo collaborativo) e consegnare alcuni esercizi, la cui correzione e valutazione verrà effettuata dal tutore della sua classe e consegnata individualmente all'alunno durante la settimana successiva. Oltre al lavoro individuale sono previste forme di apprendimento collaborativo attraverso forum di discussione in ciascuna classe, e (almeno un) forum comune alle due classi, ad esempio per la discussione di problemi organizzativi comuni. Infine, il docente ed i tutori desiderano disporre di uno spazio ad essi riservato, ad esempio per discutere problemi sulla valutazione, sui materiali didattici, concordare decisioni riguardo a problemi imprevisti, etc.. Il problema è come organizzare gli spazi sul *BSCW* in modo da garantire il rispetto dei limiti di visibilità conseguenti ai requisiti suddetti. Una semplice soluzione è... già stata presentata, in forma astratta e semplificata, con il DAG di fig. 2.2.

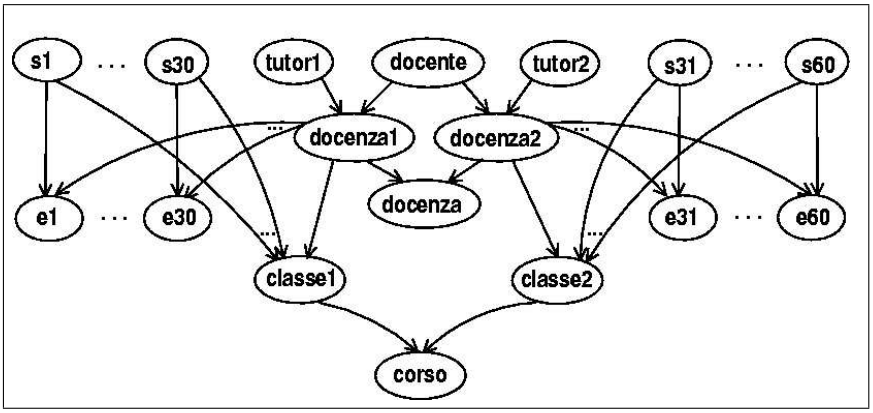

Figura 2.3. DAG di gruppi *BSCW* per un corso in rete con due classi

Per rendersi conto di come tale DAG contenga l'essenza della soluzione dovrebbe essere sufficiente confrontarlo con il DAG in fig. 2.3.

L'esempio bene illustra il perché in un DAG di gruppi *BSCW* siano quasi sempre vere *entrambe* le ragioni che impediscono al DAG di essere un albero: 1) come detto sopra, ogni utente del sistema costituisce un gruppo individuale, che è un nodo privo di predecessori nel DAG dei gruppi; 2) ogniqualvolta due o più gruppi (individuali o meno) sono invitati a condividere uno spazio, essi divengono *i predecessori* (più d'uno) del gruppo associato allo spazio.

Quanto detto chiarisce il significato della relazione di *predecessore* rappresentata dagli archi del DAG: un arco da G a H significa che il gruppo G è membro del gruppo H, dunque ogni utente in G ha accesso allo spazio associato ad H, e infatti lo spazio associato al gruppo H è un sottospazio di quello associato al gruppo G, ovvero è visibile come cartella in quest'ultimo.

Un DAG *BSCW* rende efficacemente il quadro di chi ha accesso a quali spazi: un utente ha accesso ad uno spazio se e solo se nel DAG esiste un *cammino*dal suo gruppo individuale al gruppo associato allo spazio. Avere accesso ad uno spazio *non* equivale dunque ad essere membro del gruppo associato a quello spazio: questo equivale piuttosto all'esistenza, nel DAG, di un cammino costituito da un solo arco.

Un utente può avere accesso ad uno spazio in più modi diversi, ovvero, nel DAG esistono più cammini dal suo gruppo individuale al gruppo associato allo spazio in questione. Questo fatto ha due importanti conseguenze pratiche.

1. Rimuovere un (gruppo) membro da un gruppo non necessariamente significa togliere ai suoi utenti l'accesso allo spazio associato al gruppo; tale operazione si traduce nella rimozione di un arco dal DAG, ma possono esistere altri cammini dai gruppi individuali degli utenti in questione al gruppo a cui puntava l'arco rimosso.
2. Ogni invito di un nuovo membro a condividere uno spazio gli assegna un *ruolo*, che ne determina i diritti di accesso; vale inoltre l'*ereditarietà dei ruoli* lungo i cammini, coerentemente con l'osservazione fatta alla fine della precedente sez. 2.2. Dunque un utente può ben avere accesso ad uno spazio con più ruoli diversi, corrispondenti a diversi cammini dal suo gruppo individuale al gruppo associato allo spazio in questione. La rimozione di un arco dal DAG può dunque, in certe circostanze, comportare una modifica dei diritti di accesso di alcuni utenti allo spazio del gruppo a cui puntava l'arco rimosso. Ci si potrebbe aspettare che tale modifica possa solo essere una *perdita* di alcuni (o tutti) i diritti di accesso, ma la prossima sez. 2.4 riserva qualche sorpresa a chi nutre tale aspettativa.

2.4 La negazione in modelli di autorizzazione

Quando un utente ha accesso ad uno spazio o altro oggetto con un solo ruolo, questo definisce quali diritti di accesso sono autorizzati per l'utente e quali non lo sono. Nella precedente sez. 2.3 si è però visto che un utente può accedere ad uno spazio con più ruoli, attraverso cammini diversi nel DAG

dei gruppi. Questo comporta anche una possibile molteplicità di ruoli che l'utente può avere sugli oggetti contenuti nello spazio (per l'ereditarietà dei ruoli). Inoltre, e indipendentemente dalle ragioni dette, è possibile assegnare nuovi ruoli, in aggiunta a quelli già assegnati, ad utenti che vi hanno accesso, attraverso una operazione definita a tal scopo. Ora, supponiamo che un certo diritto di accesso, ad esempio quello di modifica, sia attribuito ad un utente attraverso un ruolo e non attribuito attraverso un altro. Qual è l'attribuzione risultante in tal caso? Questa domanda ha risposte diverse in modelli di autorizzazione diversi.

Nei modelli di autorizzazione *disgiuntiva*, il soggetto gode di un diritto di accesso su un oggetto se tale diritto gli è attribuito in *almeno uno* dei ruoli che gli sono assegnati su quell'oggetto. Viceversa, nei modelli di autorizzazione *congiuntiva*, si richiede che il diritto in questione sia attribuito al soggetto in *tutti* i ruoli che gli sono assegnati per quell'oggetto. Nell'autorizzazione disgiuntiva, dunque, la presenza del diritto di accesso in uno qualsiasi dei ruoli assegnati al soggetto funziona implicitamente da attribuzione di tale diritto *comunque*, mentre nell'autorizzazione congiuntiva è l'*assenza* del diritto in uno qualsiasi dei ruoli assegnati che funziona da *negazione* dell'esercizio di tale diritto *comunque*. Entrambi gli approcci presentano inconvenienti pratici facilmente intuibili: questo fatto ha motivato la ricerca di modelli in cui uno dei due approcci sia quello di *default*, ma sia anche possibile far valere un criterio diverso in casi specifici, quando lo si desideri, attraverso un esplicito meccanismo a tal scopo. Vediamo brevemente alcuni casi significativi.

Il concetto di *diritto di accesso negativo* è adoperato nel sistema *Andrew* [3], che è disgiuntivo rispetto agli ordinari diritti di accesso (positivi), ma prevede anche gli stessi diritti con segno opposto, cioè *divieti*. Quando un diritto di accesso è attribuito con entrambi i segni, prevale il segno negativo. Questo modello è semplice da realizzare, ma non sempre risulta adeguato alle esigenze pratiche. Più flessibile è l'approccio seguito nel modello proposto in [4], dove non c'è una prevalenza *a priori* del divieto, bensì altri fattori determinano, in caso di conflitto, quale dei due segni debba prevalere. L'idea è buona in linea di principio, ma complica notevolmente la realizzazione, in quanto si richiede l'introduzione di parecchie regole di risoluzione di conflitti per decidere se l'autorizzazione va concessa o meno in tali casi. Un concetto prossimo a quello di diritto di accesso negativo, ma diverso, è stato proposto in [5], ed applicato nelle versioni 3 del sistema *BSCW*: l'*esclusione* di un utente da un gruppo, o altrimenti detta *appartenenza negativa* del suo gruppo individuale ad un gruppo, vieta che l'utente possa avere accesso allo spazio associato al gruppo, indipendentemente dal conferimento dell'accesso per altra via.

La suddetta forma di diritto di accesso negativo è stata sostituita nella versione 4 del *BSCW* dall'introduzione, in un modello peraltro disgiuntivo, del ruolo predefinito di *Membro ristretto*, dotato di una peculiarità negativa che lo distingue dagli altri ruoli predefiniti: i diritti di accesso non attribuiti

da questo ruolo sono negati, indipendentemente dall'attribuzione che se ne può avere attraverso altri ruoli. Il ruolo di *Membro ristretto* è l'unico *fra i ruoli predefiniti* ad avere questo "potere di veto", ma è ben possibile definire altri ruoli dotati dello stesso potere: a tale scopo occorre semplicemente che, al momento della loro prima creazione, essi vengano definiti usando il ruolo *Membro ristretto* come modello.

È invece chiaramente raccomandata dai manuali, e confermata dall'esperienza, la massima cautela con l'uso dei ruoli predefiniti nel *BSCW*. Particolarmente pericolosa per applicazioni alla didattica in rete, in quanto fonte probabile di sgradevoli sorprese, è la combinazione dei due seguenti *default*:

1. quando si invita qualcuno a condividere uno spazio, il *ruolo* che gli si assegna per default è quello di *Membro*;
2. i *tipi di diritti di accesso* di cui consta il ruolo predefinito di *Membro* sono tutti, tranne uno che riguarda operazioni di delega (a cui accenniamo nella prossima sez. 2.5).

Una delle conseguenze della combinazione "automatica" di questi default è che, ad esempio, se si invitano gli studenti di una classe ad uno spazio comune con il ruolo di *Membro*, senza aver preventivamente ridotto i diritti di accesso che questo conferisce, nello spazio in questione tutti avranno il diritto di 1) cambiar nome allo spazio, 2) cambiarne la descrizione, 3) rimuovere qualsiasi oggetto dallo spazio, compresi sottospazi, 4) rimuovere membri dal gruppo associato allo spazio, ...

L'esperienza insegna che, oltre ai diritti di lettura, i diritti di *aggiunta* sono quelli che servono, mentre quelli di *modifica* e *rimozione* comportano i rischi suddetti, e sono dunque da negare. Questo non impedirà agli utenti che hanno accesso allo spazio di modificare o rimuovere gli oggetti di *propria* creazione, in quanto il ruolo assegnato per default a chi crea un oggetto gli attribuisce tali diritti su di esso.

2.5 Autorizzazione e delega

Il raggruppamento di diritti di accesso in tipi e in ruoli, visto in sez. 2.2, e quello di utenti in gruppi, secondo le strutture viste in sez. 2.3, naturalmente fanno pensare a questi insiemi e strutture come oggetti su cui si può agire attraverso operazioni. In effetti è così, ma allora si presenta immediatamente la questione dell'autorizzazione anche per tali oggetti ed operazioni: ad esempio, se la definizione dei ruoli per un dato oggetto è essa stessa un oggetto, dovrà avere una ACL associata che stabilisca chi è autorizzato a compiere quali operazioni su tale definizione dei ruoli. Fin qui la cosa ha un senso, ma questo comincia a diventare piuttosto evanescente se si consente la reiterazione di questo meccanismo *ad libitum*: nel caso in questione, ciò significherebbe contemplare la ACL associata ad una definizione di ruoli come un oggetto essa stessa, con una sua *propria* definizione di ruoli ed ACL associata

a quest'ultima, donde trae origine una catena di concetti che ha lunghezza illimitata ma dubbia utilità pratica.

L'arresto di catene quali quella considerata sopra al secondo livello corrisponde ad un analogo arresto all'autoapplicazione del concetto di autorizzazione: l'autorizzazione all'autorizzazione è un caso particolare del concetto di *delega*, ovvero conferimento ad altri di diritti di accesso e delle associate responsabilità di cui si è titolari, senza necessariamente perderli per questo, ed eventualmente con possibilità di *revoca* della delega. La delega occorre di frequente negli ambienti collaborativi. L'arresto in questione può essere realizzato, come nel caso del *BSCW* [5], distinguendo due categorie di oggetti:

- oggetti *regolari*: a ciascuno di essi è associata una ACL che autorizza l'accesso degli utenti alle operazioni su di essi attraverso la definizione ed assegnazione di ruoli;
- oggetti *attributi*: a differenza dei primi, un oggetto di questa categoria non ha ACL propria, ma è piuttosto un'entità associata ad un oggetto regolare, a cui si accede con operazioni definite per l'oggetto regolare (dunque secondo l'autorizzazione definita nella ACL dell'oggetto regolare), e che cessa di esistere quando l'oggetto regolare cessa di esistere.

La *descrizione* associata ad un oggetto (regolare) è un comune esempio di oggetto attributo nel *BSCW*. La definizione dei ruoli per un oggetto (regolare) e l'assegnazione di ruoli ad utenti per esso sono esempi di oggetti attributi rilevanti al concetto di delega. Due tipi di diritti di accesso sono rilevanti alla delega nel *BSCW*: la *condivisione estesa*, in cui sono classificate le operazioni di invito all'accesso e revoca dell'invito, e il tipo *Aux 1* (in cerca di un nome migliore), per le operazioni di definizione e modifica di ruoli, e assegnazione di ruoli ad utenti. Quest'ultima operazione può essere adoperata per realizzare la delega ricorsiva (delega alla delega), semplicemente includendo il diritto ad eseguire tale operazione fra i diritti di accesso del ruolo che si assegna.

Alla fine della precedente sez. 2.4 si è accennato ai rischi derivanti dal conferimento automatico del ruolo predefinito di *Membro* di uno spazio comune agli studenti di una classe: tutti sono autorizzati a cancellare tutto, ivi compresa la cancellazione di membri dal gruppo associato allo spazio, giacché l'unico tipo di diritti di accesso escluso da questo ruolo è il tipo *Aux 1* detto sopra. Esaminiamo ora una situazione in cui questa esclusione può risultare problematica. Supponiamo che il docente del corso illustrato in fig. 2.3 inviti i due tutori assegnando loro il ruolo predefinito di *Membro*, e chieda loro di organizzare i rispettivi spazi delle classi. Ciascun tutore potrà creare lo spazio di sua competenza e invitare gli studenti della sua classe, ma non sarà abilitato a modificare i ruoli predefiniti né a definirne di nuovi: la delega ricevuta, troppo ampia per essere conferita agli studenti, risulta troppo limitata per le ordinarie esigenze di conduzione di una classe on-line.

Bibliografia

1. *Basic Support for Cooperative Work*, `http://bscw.gmd.de`
2. Lampson, B.W., Protection, *ACM Operating Systems Review*, **8** (1974) 18–24.
3. Satyanarayanan, M., Integrating Security in a Large Distributed System, *ACM Transactions on Computer Systems*, **7** (1989) 247–280.
4. Shen, H. and Dewan, P., Access Control for Collaborative Environments, in: *Proc. ACM Conference on Computer-Supported Cooperative Work (CSCW'92)*, Toronto, Canada (1992) 51–58.
5. Sikkel, K., A Group-Based Authorization Model for Cooperative Systems, in: *Proc. European Conference on Computer-Supported Cooperative Work (ECSCW'97)*, Lancaster, September 1997; Kluwer, Dordrecht, NL (1997).
6. *Zope Community*, `http://www.zope.org`