

**Università degli Studi di Pisa**  
**Facoltà di Scienze Matematiche Fisiche e Naturali**  
**Corso di laurea in Informatica**

**Sistemi di Elaborazione dell'informazione:  
Gestione di reti  
prof. Luca Deri**

**Definizione di un MIB SNMP per il controllo di software  
applicativo**

**Maria Teti**  
<*teti@cli.di.unipi.it*>

Anno Accademico: 2001/2002 - Sessione estiva

# 1- Introduzione

Le applicazioni software, oggi, costituiscono uno degli elementi più importanti presenti su un pc. Per le tali applicazioni importante risulta la gestione, anche perché mantenere il controllo sulle applicazioni permette di rendere più efficiente l'uso del pc da parte degli utenti, evitando un spreco di risorse.

Al giorno d'oggi i sistemi operativi, come Linux e Windows, presenti sul mercato, permettono la gestione del software installato tramite una interfaccia grafica (GUI).

Nella figura viene mostrato la GUI relativa al sistema Windows.

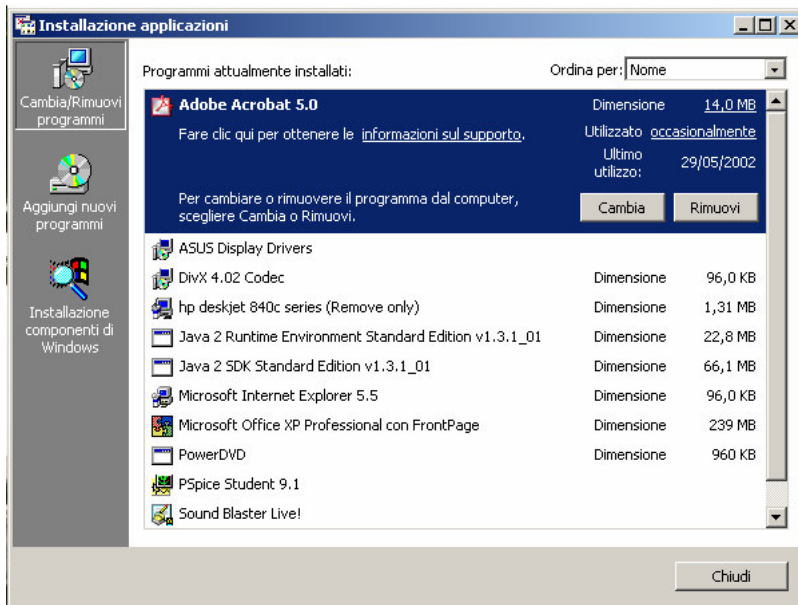


figura 1

La gestione degli applicativi può riguardare ad esempio tre aree funzionali importanti:

- sicurezza
- configurazione
- fault.

Per sicurezza si intende per prima cosa capire quali applicazioni vengono installate con una licenza valida e quali invece risultano essere copie pirata. Significa inoltre poter stabilire quali siano i diritti di accesso alle applicazioni da parte di singoli utenti o gruppi.

Per configurazione si intende lo spazio occupato dalle cartelle e sottocartelle che costituiscono l'applicazione, il path su cui è installata l'applicazione e così via.

Per fault si intende invece stabilire se l'applicazione ha funzionato correttamente o meno.

Nell'interfaccia di Windows troviamo anche informazioni di accounting come la data di ultimo utilizzo dell'applicazione.

Lo scopo di questo progetto è quello di effettuare una gestione del software installato utilizzando il protocollo SNMP (Simple Network Management Protocol). Il protocollo SNMP è un protocollo di tipo manager-agent, dove sia per agent che per il manager si intende un programma software. L'agent risiede solitamente nella risorsa che deve essere gestita e nel caso affrontato in questo progetto risiede nel pc dove è installato il software applicativo. Il manager invece non risiede sullo stesso pc (il fatto che il manager non risiede sulla risorsa che deve essere gestita è tipico di questo protocollo). SNMP è proprio il protocollo utilizzato per far comunicare tra di loro l'agent e il manager. In realtà quella che viene gestita non è la risorsa reale, ma una visione astratta della stessa, che rappresenta solo le caratteristiche che interessano per la gestione. Si tratta dei Managed Object che vengono raggruppati nei MIB. I valori di questi oggetti possono essere richiesti o settati dal manager tramite particolari messaggi SNMP.

## **2. Soluzione proposta**

Il problema di gestire un insieme di applicazioni installate su un host o su un pc è stato affrontato in differenti MIB che hanno proposto soluzioni differenti e in maniera più o meno dettagliata e approfondita a seconda del MIB considerato. Nella RFC 2790, che definisce l' Host Resource Mib, vengono gestite le applicazioni che stanno girando e quelle che sono caricate in memoria fisica o virtuale pronte per essere inviate in esecuzione. Vengono inclusi nella gestione anche il sistema operativo della macchina, i driver dei device e le applicazioni software in generale. Nella RFC 2287 vengono considerati i pacchetti applicativi e gli elementi individuali che compongono una applicazione (file ed eseguibili). Inoltre vengono gestite le applicazioni che sono già andate in esecuzione sulla macchina.

Quello che si vuole fare con la scrittura di questo MIB è implementare una politica di gestione simile a quella proposta da Windows nel suo pannello “applicazioni installate”, integrando nelle parti che sono sembrate poco trattate, ma che sono utili per l’amministratore.

Il MIB si compone di due gruppi di oggetti, *applObjects* e *applTrap*. Il primo gruppo contiene due tabelle, una per i pacchetti software installati e uno per le singole applicazioni installate sulla macchina. Molti degli attributi della tabelle sono simili. La tabella per le singole applicazioni è quella che contiene la informazioni più importanti per la gestione, mentre quella relativa ai pacchetti ha solo lo scopo di dare informazioni riassuntive e un visione generale per le applicazioni che appartengono ad un pacchetto software.

Contiene anche una soglia che permette di controllare lo spazio massimo che può esser occupato da tutte la applicazioni installate, in modo da poter capire se si sta occupando troppo spazio sul disco, rendendo poco efficiente l’utilizzo di questa risorsa per altri scopi.

Il controllo sulla quantità di spazio occupabile viene fatto con gli attributi *applMaxDisc*, che indica la quantità massima di disco che è possibile occupare per le applicazioni, e *applDiscUsed*, che indica la quantità di disco realmente occupata dalle applicazioni installate. Tale quantità è la somma dello spazio occupato da tutte le applicazioni subito dopo l’installazione. Se *applDiscUsed* supera il valore di *applMaxDisc* viene inviata una trap al manager ( viene inviata una trap anche se si rientra sotto tale soglia).

Un altro controllo che si è scelto di effettuare con questo MIB è se la applicazioni, quando vengono invocate, sono realmente utilizzate o meno. Per effettuare questo tipo di controllo si utilizzano due attributi presenti nella tabella *applCompTable*, *applCompInvocTime* e *applCompRealTime*. Il primo attributo indica il tempo totale di tutte le invocazioni di una applicazione in una giornata, mentre il secondo indica il tempo di utilizzo totale (calcolato dal valore dell’attributo precedente meno i secondi del timer dello screen saver). Se il tempo reale e’ minore della metà del tempo di utilizzo allora si invia una trap al manager. La trap viene inviata ogni volta che i valori *applCompRealTime* e *applCompInvocTime* vengono modificati e si verifica la condizione specificata prima.

Gli oggetti del primo gruppo sono:

- **applMaxDisc:** E’ la quantità di disco massima, in kbyte, che può esser occupata dall’installazione di tutte le applicazioni. Superata tale soglia si invia una trap. Ciò avviene anche quando si rientra sotto tale valore.
- **applDiscUsed:** E’ la quantità di disco, in kbyte, usata da tutte le applicazioni al momento dell’installazione (la somma della quantità di disco occupata da tutte le applicazioni dopo l’installazione). Se questo valore supera il valore di *applMaxDisc* viene inviata una trap al manager. Un’altra trap viene inviata anche quando si scende nuovamente al di sotto di tale valore.

Gli attributi della prima tabella, *applPkgTable*, sono:

- **applPkgIndex**: Indice della tabella che serve per identificare il pacchetto installato. Viene incrementato di uno ogni volta che viene installato un nuovo pacchetto software;
- **applPkgNome**: Indica il nome del pacchetto software installato;
- **applPkgManufacturer**: E' il nome del costruttore del pacchetto software;
- **applPkgVersion**: E' il numero della versione assegnato dal costruttore al pacchetto. Ogni volta che un pacchetto viene aggiornato con una nuova versione, viene inviata una trap contenente il numero della nuova versione e la data di aggiornamento;
- **applPkgLicense**: Indica se si tratta di un pacchetto open source o se si tratta di un pacchetto che ha bisogno di una licenza d'uso.
- **applPkgSerialNumber**: E' il numero seriale assegnato dal costruttore al pacchetto. Ogni volta che un nuovo pacchetto viene installato sulla macchina, viene inviata al manager una trap contenente il numero seriale per una verifica della licenza d'uso se il pacchetto non è open source;
- **applPkgOwner**: Indica il nome del proprietario registrato durante la fase di installazione. Se non viene registrato nessun proprietario questo campo è vuoto.
- **applPkgInstDate**: E' la data di installazione del pacchetto.
- **applPkgModDate**: E' la data di modifica o aggiornamento del pacchetto, dove per modifica si intende l'aggiunta di nuove applicazioni che fanno comunque parte del pacchetto stesso; Può essere utile per determinare se un pacchetto è molto vecchio e debba essere aggiornato;
- **applPkgSpace**: E' lo spazio su disco occupato dal pacchetto subito dopo l'installazione;
- **applPkgDate**: E' la data di ultimo utilizzo del pacchetto. Può essere utile per stabilire se il pacchetto risulta inutilizzato da tanto tempo e possa essere quindi eliminato. Viene riportata la data di ultimo utilizzo dell'applicazione appartenente al pacchetto che è stata invocata per ultima;
- **applPkgCrash**: E' il numero di crash a cui il pacchetto è andato incontro. E' la somma di tutti i crash subiti dalle applicazioni che lo compongono; Tale calcolo è possibile sulle applicazioni non open source perché le applicazioni che fanno parte di uno stesso pacchetto hanno tutte lo stesso numero seriale.

Gli attributi della seconda tabella, *applCompTable*, sono i seguenti:

- **applCompIndex**: E' uno degli indici della tabella che serve per identificare le applicazioni installate. Viene incrementato di uno ogni volta che viene installata una nuova applicazione;
- **applCompName**: E' il nome assegnato dal costruttore all'applicazione installata;
- **applCompPkgIndex**: E' il secondo indice per questa tabella. Indica il pacchetto di cui l'applicazione fa parte. Il suo valore è quello dell'indice *applPkgIndex* contenuto nella tabella *applPkgTable*;
- **applCompManufacturer**: E' il nome del costruttore dell'applicazione;

- **applCompVersion:** E' il numero della versione dell'applicazione installata. E' assegnato dal costruttore. Ad ogni aggiornamento viene inviata un trap contenete il numero della nuova versione;
- **applCompLicense:** Indica se l'applicazione è open source o no. Viene considerato per le singole applicazioni perché si potrebbero installare delle applicazioni che non fanno parte di nessun pacchetto applicativo.
- **applCompSerialNumber:** E' il numero seriale dell'applicazione. Proprio come per il pacchetto serve per poter controllare se la licenza d'uso è valida nel coso in cui l'applicazione non è open source;
- **applCompOwner:** Indica il proprietario dell'applicazione registrato durante la fase di installazione. Se il proprietario non è stato registrato questo campo è vuoto.
- **applCompPath:** E' il path assoluto su cui è installata l'applicazione;
- **applCompInsDate:** E' la data di installazione dell'applicazione;
- **applCompModDate:** E' la data di aggiornamento dell'applicazione;
- **applCompSpace:** Spazio occupato dall'applicazione subito dopo l'installazione;
- **applCompDate:**E' la data di ultima invocazione dell'applicazione;
- **applCompCrash:** Numero di crash a cui l'applicazione è andata incontro;
- **applCompUsers:** Nome dell'ultimo utente che ha invocato l'applicazione;
- **applCompUseTime:** Permette di stabilire la frequenza con cui viene chiamata l'applicazione in base al numero di invocazioni da parte degli utenti ogni settimana. Alla fine della settimana viene inviata una trap se l'applicazione non è stata mai invocata.
- **applCompInvocTime:** E' il tempo in secondi della durata di tutte le esecuzioni di una applicazione avvenute in una giornata;
- **applCompRealTime:** E' il tempo di effettivo utilizzo dell'applicazione durante l'applCompInvocTime. Questo tempo può essere calcolato solo per quelle applicazioni che richiedono in maniera preponderante l'uso interattivo di tastiera e mouse come per esempio gli editor di testi. Ciò serve per capire se una applicazione viene realmente utilizzata. Si calcola sottraendo ad applCompInvocTime il tempo del timer dello screen-saver. Se questo valore è inferiore alla metà di applCompInvocTime viene inviata una trap (esattamente viene inviata quando di aggiorna il valore di applCompInvocTime e applCompRealTime e si verifica la condizione specificata prima).

Il secondo gruppo contiene le trap definite:

- **installazioneApplicazione:** Viene inviata una nuova trap di questo tipo ogni volta che una nuova applicazione , appartenente o meno ad un pacchetto, viene installata.
- **faultApplicazione:** Questo tipo di trap viene inviata ogni volte che una applicazione, appartenente o meno ad un pacchetto, va incontro ad un crash. Se l'applicazione appartiene ad un pacchetto si va ad incrementare il numero di crash dell'intero pacchetto.
- **disinstallazioneApplicazione:** Viene invita quando una applicazione, appartenente a un pacchetto o meno, viene disinstallata.
- **aggiornamentoApplicazione:** Viene inviata quando una applicazione, appartenente ad un pacchetto o meno, viene aggiornata con una nuova versione.

- **installazionePacchetto:** Come per le applicazioni viene inviata una trap ogni volta che viene installato un nuovo pacchetto applicativo sul pc.
- **aggiornamentoPacchetto:** Viene inviata quando un intero pacchetto applicativo viene aggiornato con una nuova versione.
- **disinstallazionePacchetto:** Come per le applicazioni viene inviata una trap ogni volta che viene disinstallato un intero pacchetto applicativo.
- **frequenzaMaiApplicazione:** E' una trap inviata al manager quando per una intera settimana una applicazione non viene invocata dagli utenti.
- **maxdiscApplicazioni:** E' una trap che viene inviata ogni volta che il valore di applDiscUsed supera quello della soglia applMaxDisc. La stessa trap viene inviata ogni volta che si scende sotto tale valore. Questa trap serve per evitare di occupare troppo spazio su disco fisso per l'installazione di software applicativo.
- **realtimeApplicazioni:** Viene inviata ogni volta che il tempo di utilizzo reale dell'applicazione (applCompRealTime) scende al di sotto del valore della metà di applCompInvocTime. Viene inviata esattamente quando il valore di questi due attributi viene aggiornato e si verifica la condizione specificata prima.

### 3. II MIB

```
MYAPPL-MIB DEFINITIONS ::= BEGIN
```

#### IMPORTS

```
MODULE-IDENTITY, OBJECT-TYPE, enterprises,
Integer32, Counter32, Unsigned32,
TimeTricks, Gauge32                                FROM SNMPv2-SMI
```

```
TEXTUAL-CONVENTION, DisplayString,
DateAndTime                                        FROM SNMPv2-TC
```

```
Kbytes                                            FROM HOST-RESOURCE-MIB
```

```
myApplMib MODULE-IDENTITY
LAST-UPDATED "200206142329z"
ORGANIZATION "Maria Teti Inc."
```

**CONTACT-INFO**

"Maria Teti  
Universita' degli Studi di Pisa  
Pisa, Italy  
e-mail: [teti@cli.di.unipi.it](mailto:teti@cli.di.unipi.it)"

**DESCRIPTION**

"Modulo MIB per la gestione di software applicativo  
installato su un pc"

**REVISION** "200206142329z"

**DESCRIPTION**

"Sono state modificate la tabelle definite nel Mib,  
aggiungendo degli attributi e spiegando come tali  
attributi possono essere calcolati. Inoltre sono state  
inserite delle soglie e sono state inserite delle  
nuove trap per il superamento di tali soglie."

**REVISION** "200206091004z"

**DESCRIPTION**

"Versione originale del MIB"  
 ::= { private 67 }

-- gruppi di oggetti

applObjects OBJECT IDENTIFIER ::= { myApplMib 1 }

applTrap OBJECT IDENTIFIER ::= { myApplMib 2 }

-- definizioni textual convention

Frequency ::= **TEXTUAL-CONVENTION**

**STATUS** current

**DESCRIPTION**

"Convenzione testuale che descrive la frequenza con  
cui le applicazioni vengo invocata settimanalmente"

**SYNTAX** **INTEGER** {  
high (1), medium (2), low (3), never(4)  
}

License ::= **TEXTUAL-CONVENTION**

**STATUS** current

**DESCRIPTION**



"Convenzione testuale che descrive se una applicazione  
 ha bisogno di una licenza d'uso o se è open-source"  
**SYNTAX**            **INTEGER** {  
                                  license (1), open-source (2)  
                                  }

-- definizione oggetti

applMaxDisc **OBJECT-TYPE**

**SYNTAX**            Unsigned32

**UNITS**             "Kbytes"

**MAX-ACCESS**      read-write

**STATUS**            current

**DESCRIPTION**

"E' la quantità di disco massima, in kbyte, che può  
 esser occupata dall'installazione di tutte le  
 applicazioni. Superata tale soglia si invia una trap.  
 Ciò avviene anche quando si rientra sotto tale  
 valore."

::= { applObjects 1 }

applDiscUsed **OBJECT-TYPE**

**SYNTAX**            Unsigned32

**UNITS**             "Kbytes"

**MAX-ACCESS**      read-only

**STATUS**            current

**DESCRIPTION**

"E' la somma delle quantità di disco, in kbyte,  
 occupato da tutte le applicazioni al momento  
 dell'installazione. Se questo valore supera il valore  
 di applMaxDisc viene inviata una trap al manager.  
 Un'altra trap viene inviata anche quando si scende  
 nuovamente al di sotto di tale valore."

::= { applObjects 2 }

applPkgTable **OBJECT-TYPE**

**SYNTAX**            **SEQUENCE OF** ApplPkgEntry

**MAX-ACCESS**      not-accessible

**STATUS**            current

**DESCRIPTION**

“Tabella (concettuale) contenente tutti i pacchetti software installati sulla macchina”

::= { applObjects 3 }

**applPkgEntry OBJECT-TYPE**

**SYNTAX** ApplPkgEntry

**MAX-ACCESS** not-accessible

**STATUS** current

**DESCRIPTION**

“Entrata (concettuale) che descrive un pacchetto applicativo installato”

**INDEX** { applPkgIndex }

::= { applPkgTable 1 }

```
applPkgEntry ::= SEQUENCE {
    applPkgIndex          Integer32,
    applPkgName           DisplayString,
    applPkgManufacturer   DisplayString,
    applPkgVersion        DisplayString,
    applPkgLicense        License,
    applPkgSerialNumber   DisplayString,
    applPkgOwner           DisplayString,
    applPkgInsDate         DateAndTime,
    applPkgModDate         DateAndTime,
    applPkgSpace           Kbytes,
    applPkgDate            DateAndTime,
    applPkgCrash           Counter32,
}
```

**applPkgIndex OBJECT-TYPE**

**SYNTAX** Integer32

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Intero utilizzato come indice. Viene incrementato di uno ogni volta che viene installato un nuovo pacchetto software”

::= { applPkgEntry 1 }

applPkgName **OBJECT-TYPE**

**SYNTAX** DisplayString

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Nome del pacchetto software installato”

::= { applPkgEntry 2 }

applPkgManufacturer **OBJECT-TYPE**

**SYNTAX** DisplayString

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Costruttore del pacchetto software installato”

::= { applPkgEntry 3 }

applPkgVersion **OBJECT-TYPE**

**SYNTAX** DisplayString

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Numero versione assegnato dal costruttore al pacchetto”

::= { applPkgEntry 4 }

applPkgLicense **OBJECT-TYPE**

**SYNTAX** License

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Indica se si tratta di un pacchetto open source o se si tratta di un pacchetto che ha bisogno di una licenza d'uso.”

::= { applPkgEntry 5 }

applPkgSerialNumber **OBJECT-TYPE**

**SYNTAX** DisplayString

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Numero seriale assegnato dal costruttore al

```

        pacchetto"
    ::= { applPkgEntry 6 }

applPkgOwner OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " "
    ::= { applPkgEntry 7 }

applPkgInsDate OBJECT-TYPE
    SYNTAX          DateAndTime
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Data di installazione del pacchetto software "
    ::= { applPkgEntry 8 }

applPkgModDate OBJECT-TYPE
    SYNTAX          DateAndTime
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Data di modifica o aggiornamento del pacchetto
        software "
    ::= { applPkgEntry 9 }

applPkgSpace OBJECT-TYPE
    SYNTAX          Kbytes
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Spazio sul disco occupata del pacchetto subito dopo
        l'installazione "
    ::= { applPkgEntry 10 }

applPkgDate OBJECT-TYPE
    SYNTAX          DateAndTime
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION

```

"Data di ultimo utilizzo del pacchetto"  
 ::= { applPkgEntry 11 }

applPkgCrash **OBJECT-TYPE**

**SYNTAX** Counter32

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

    "Numero di crash a cui e' andato incontro il pacchetto. E' la somma di tutti i crash a cui sono andate incontro le applicazioni che compongono il pacchetto. Per le applicazioni non open source il calcolo viene fatto facilmente perché le applicazioni che appartengono ad uno stesso pacchetto hanno tutte lo stesso numero seriale."

::= { applPkgEntry 12 }

applCompTable **OBJECT-TYPE**

**SYNTAX** SEQUENCE OF ApplCompEntry

**MAX-ACCESS** not-accessible

**STATUS** current

**DESCRIPTION**

    "Tabella (concettuale) contenente tutte le applicazioni che costituiscono i pacchetti"

::= { applObjects 4 }

applCompEntry **OBJECT-TYPE**

**SYNTAX** ApplCompEntry

**MAX-ACCESS** not-accessible

**STATUS** current

**DESCRIPTION**

    "Entrata (concettuale) che descrive una applicazione appartenente ad un pacchetto applicativo installato"

**INDEX** { applCompIndex }

::= { applCompTable 1 }

applCompEntry ::= SEQUENCE {

applCompIndex	Integer32,
applCompName	DisplayString,
applCompPkgIndex	Integer32,
applCompManufacturer	DisplayString,
applCompVersion	DisplayString,

applCompLicense	License,
applCompSerialNumber	DisplayString,
applCompOwner	DisplayString,
applCompPath	DisplayString,
applCompInsDate	DateAndTime,
applCompModDate	DateAndTime,
applCompSpace	Kbytes,
applCompDate	DateAndTime,
applCompCrash	Counter32,
applCompUser	DisplayString,
applCompUseTime	Frequency,
applCompInvocTime	DateAndTime,
applCompRealTime	DateAndTime

}

applCompIndex **OBJECT-TYPE**

**SYNTAX** Integer32

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

"Intero utilizzato come parte di indice di questa tabella. Viene incrementato di uno ogni volta che viene installata una applicazione appartenente ad un pacchetto"  
 ::= { applCompEntry 1}

applCompName **OBJECT-TYPE**

**SYNTAX** DisplayString

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

"Nome con cui e' conosciuta l'applicazione appartenente al pacchetto"  
 ::= { applCompEntry 2}

applCompPkgIndex **OBJECT-TYPE**

**SYNTAX** Integer32

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

"E' il secondo indice per questa tabella. Indica il pacchetto di cui l'applicazione fa parte. Il suo valore e' quello dell'indice applPkgIndex contenuto nella

```
    tabella applPkgTable"  
 ::= { applCompEntry 3 }
```

applCompManufacturer **OBJECT-TYPE**

**SYNTAX** DisplayString

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Costruttore dell’applicazione software installata”

```
 ::= { applCompEntry 4 }
```

applCompVersion **OBJECT-TYPE**

**SYNTAX** DisplayString

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Numero versione assegnato dal costruttore alla  
 applicazione software ”

```
 ::= { applPkgEntry 5 }
```

applCompLicense **OBJECT-TYPE**

**SYNTAX** License

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Indica se l’applicazione è open source o no. Viene  
 considerato per le singole applicazioni perché si  
 potrebbero installare delle applicazioni che non fanno  
 parte di nessun pacchetto applicativo.”

```
 ::= { applPkgEntry 6 }
```

applCompSerialNumber **OBJECT-TYPE**

**SYNTAX** DisplayString

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

“Numero seriale assegnato dal costruttore del  
 pacchetto all’applicazione”

```
 ::= { applCompEntry 7 }
```

applCompOwner **OBJECT-TYPE**

**SYNTAX** DisplayString  
**MAX-ACCESS** read-only  
**STATUS** current  
**DESCRIPTION**  
"Indica il proprietario dell'applicazione registrato durante la fase di installazione. Se il proprietario non è stato registrato questo campo è vuoto."  
 ::= { applCompEntry 8 }

applCompPath **OBJECT-TYPE**  
**SYNTAX** DisplayString  
**MAX-ACCESS** read-only  
**STATUS** current  
**DESCRIPTION**  
"Path assoluto su cui e' installata l'applicazione"  
 ::= { applCompEntry 9 }

applCompInsDate **OBJECT-TYPE**  
**SYNTAX** DateAndTime  
**MAX-ACCESS** read-only  
**STATUS** current  
**DESCRIPTION**  
"Data di installazione dell'applicazione"  
 ::= { applCompEntry 10 }

applCompModDate **OBJECT-TYPE**  
**SYNTAX** DateAndTime  
**MAX-ACCESS** read-only  
**STATUS** current  
**DESCRIPTION**  
"Data di modifica o aggiornamento dell'applicazione"  
 ::= { applCompEntry 11 }

applCompSpace **OBJECT-TYPE**  
**SYNTAX** Kbytes  
**MAX-ACCESS** read-only  
**STATUS** current  
**DESCRIPTION**  
"Spazio sul disco occupato dall'applicazione subito dopo l'installazione"  
 ::= { applCompEntry 12 }



```

applCompDate OBJECT-TYPE
  SYNTAX          DateAndTime
  MAX-ACCESS      read-only
  STATUS          current
  DESCRIPTION
    "Data di ultima invocazione dell'applicazione"
    ::= { applCompEntry 13 }
applCompCrash OBJECT-TYPE
  SYNTAX          Counter32
  MAX-ACCESS      read-only
  STATUS          current
  DESCRIPTION
    "Numero di crash a cui e' andata incontro
    l'applicazione"
    ::= { applCompEntry 14 }

applCompUser OBJECT-TYPE
  SYNTAX          DisplayString
  MAX-ACCESS      read-only
  STATUS          current
  DESCRIPTION
    "Login dell'ultimo utente che ha invocato
    l'applicazione (es: root)"
    ::= { applCompEntry 15 }

applCompUseTime OBJECT-TYPE
  SYNTAX          Frequency
  MAX-ACCESS      read-only
  STATUS          current
  DESCRIPTION
    "Indica la frequenza settimanale con cui
    l'applicazione viene chiamata. Se al termine della
    settimana l'applicazione non è stata invocata nemmeno
    una volta viene inviata una trap"
    ::= { applCompEntry 16 }

applCompInvocTime OBJECT-TYPE
  SYNTAX          DateAndTime
  MAX-ACCESS      read-only
  STATUS          current
  DESCRIPTION

```

"E' il tempo in secondi della durata di tutte le  
esecuzioni di una applicazione avvenute in una  
giornata."  
 ::= { applCompEntry 17 }

applCompRealTime **OBJECT-TYPE**

**SYNTAX** DateAndTime

**MAX-ACCESS** read-only

**STATUS** current

**DESCRIPTION**

"E' il tempo di effettivo utilizzo dell'applicazione  
durante l'applCompInvocTime. Questo tempo può essere  
calcolato solo per quelle applicazioni che richiedono  
in maniera preponderante l'uso interattivo di tastiera  
e mouse come per esempio gli editor di testi. Ciò  
serve per capire se una applicazione viene realmente  
utilizzata. Si calcola sottraendo ad applCompInvocTime  
il tempo del timer dello screen-saver. Se questo  
valore è inferiore alla metà di applCompInvocTime  
viene inviata una trap (esattamente si invia quando  
viene soddisfatta questa condizione e i valori vengono  
aggiornati). Lo stesso se tale valore rientra nella  
norma."

::= { applCompEntry 18 }

-- definizione trap

universitadipisa OBJECT IDENTIFIER ::= { enterprises 62002 }

installazioneApplicazione **NOTIFICATION-TYPE**

**ENTERPRISE** universitadipisa

**OBJECT** { applCompIndex, applCompPkgIndex, applCompName,  
applCompSerialNumber, applCompOwner}

**STATUS** current

**DESCRIPTION**

"Trap inviata quando si installa una nuova  
applicazione appartenente ad un pacchetto o meno.  
Viene inviato anche il numero seriale  
dell'applicazione così il manager può controllare se

si tenta di usare piu' volte la stessa licenza su macchine diverse. Questa politica vale se la licenza e' unica per ogni macchina. Naturalmente se il software è open source questo controllo fallisce. "

::= { applTrap 1 }

faultApplicazione **NOTIFICATION-TYPE**

**ENTERPRISE** universitadipisa

**OBJECT** { applCompIndex, applCompPkgIndex, applCompName }

**STATUS** current

**DESCRIPTION**

"Trap inviata quando una applicazione ha un crash"

::= { applTrap 2 }

disinstallazioneApplicazione **NOTIFICATION-TYPE**

**ENTERPRISE** universitadipisa

**OBJECT** { applCompIndex, applCompName }

**STATUS** current

**DESCRIPTION**

"Trap inviata quando si disinstalla una applicazione che appartiene ad un pacchetto o meno."

::= { applTrap 3 }

aggiornamentoApplicazione **NOTIFICATION-TYPE**

**ENTERPRISE** universitadipisa

**OBJECT** { applCompIndex, applCompName, applCompPkgIndex, applCompModDate, applCompVersion }

**STATUS** current

**DESCRIPTION**

"Trap inviata quando una applicazione viene aggiornata con una versione nuova"

::= { applTrap 4 }

installazionePacchetto **NOTIFICATION-TYPE**

**ENTERPRISE** universitadipisa

**OBJECT** { applPkgIndex, applPkgName, applPkgSerialNumber, applPkgOwner }

**STATUS** current

**DESCRIPTION**

"Trap inviata quando si installa un nuovo pacchetto. Viene inviato anche il numero seriale del pacchetto cosi' il manager puo' controllare se si tenta di usare

piu' volte la stessa licenza su macchine diverse.  
Questa politica vale se la licenza e' unica per ogni  
macchina. Se il pacchetto è open source l'attributo di  
numero seriale è nullo e un controllo sulla licenza  
non viene fatto. "

```
::= { applTrap 5 }
```

aggiornamentoPacchetto **NOTIFICATION-TYPE**

**ENTERPRISE** universitadipisa

**OBJECT** { applPkgIndex, applPkgName, applPkgModDate,  
applPkgVersion }

**STATUS** current

**DESCRIPTION**

"Trap inviata quando un pacchetto viene aggiornato con  
una versione nuova"

```
::= { applTrap 6 }
```

disinstallazionePacchetto **NOTIFICATION-TYPE**

**ENTERPRISE** universitadipisa

**OBJECT** { applPkgIndex, applPkgName }

**STATUS** current

**DESCRIPTION**

"Trap che viene inviata quando un intero pacchetto  
viene disinstallato."

```
::= { applTrap 7 }
```

frequenzaMaiApplicazione **NOTIFICATION-TYPE**

**ENTERPRISE** universitadipisa

**OBJECT** { applCompIndex, applCompName, applCompPkgIndex }

**STATUS** current

**DESCRIPTION**

" Trap inviata settimanalmente quando una  
applicazione durante i sette giorni non viene mai  
invocata"

```
::= { applTrap 8 }
```

maxdiscApplicazioni **NOTIFICATION-TYPE**

**ENTERPRISE** universitadipisa

**OBJECT** { applDiscUsed, applMaxDisc }

**STATUS** current

```

DESCRIPTION
    "Trap inviata quando la quantità di disco, occupata da
    tutte le applicazioni installate (applDiscUsed) supera
    il valore della soglia applMaxDisc. La stesa trap è
    inviata quando si scende al di sotto del valore della
    soglia. "
 ::= { applTrap 9 }

realtimeApplicazioni NOTIFICATION-TYPE
    ENTERPRISE      universitadipisa
    OBJECT { applCompRealTime, applCompInvocTime }
    STATUS         current
    DESCRIPTION
        "Trap inviata quando la quantità di tempo di reale
        utilizzo in un giorno scende al di sotto della metà
        del tempo totale di invocazione in una intera
        giornata. Può servire per capire l'effettivo utilizzo
        di ogni applicazione. Viene invitata quando il valore
        dei due oggetti inviati nella trap cambia e il valore
        di applCompRealTime scende al di sotto della metà del
        valore di applCompInvocTime."
 ::= { applTrap 10 }

END

```

## 4. Conclusioni

Il controllo sullo spazio occupato dalle applicazioni installate sulla macchina poteva essere affrontato in maniera differente. Per esempio, applDiscUsed è stato definito come lo spazio totale occupato da tutte le applicazioni subito dopo l'installazione. Un controllo più accurato poteva essere fatto considerando sempre lo spazio massimo che le applicazioni possono occupare sul disco, e invece del valore di applDiscUsed definito in questo MIB, si poteva considerare una stima mensile dello spazio occupato dalle applicazioni, considerando cartelle e sottocartelle. Tale controllo diventa però più difficile a livello di implementazione.

Inoltre la soglia definita per il controllo dell'utilizzo giornaliero di una applicazione, non solo potrebbe essere aumentata, ma potrebbe essere differenziata a seconda delle applicazioni installate sulla macchina. Permettendo così una stima sull'utilizzo in maniera differente.

All'interno del MIB, infine, è stata poco trattata la problematica legata alla performance delle applicazioni. Potevano essere considerati degli attributi come il carico medio di CPU durante l'ultima invocazione (e il carico medio di RAM), ma tale tipo di informazione è dinamica, e quindi difficile da poter esser gestita.

## 5. Riferimenti

- [1] *SNMP*: <http://www.snmp.org>
- [2] *IETF*: <http://www.ietf.org>
- [3] P. Grillo, S. Waldbusser. *Host Resource Mib*, RFC 1514, Settembre 1993.
- [4] P. Grillo, S. Waldbusser. *Host Resource Mib*, RFC 2790, Marzo 2000.
- [5] C. Krupczak, J. Saperia. *Definitions of System-Level Managed Objects for Applications*, RFC 2287, Febbraio 1998.
- [6] J. Schonwalder, L. Deri. *Sistemi di Elaborazione dell'Informazione: Gestione di Rete*.