UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Facoltà di Ingegneria - Sede di Modena
Corso di Laurea Specialistica in Ingegneria Informatica

Progetto e realizzazione di un'architettura di gestione per dati personali e autenticazione

Relatore:
Chiar. ma Prof. Sonia Bergamaschi

Tesi di Laurea di:
Generali Matteo

Anno Accademico 2005/2006

Il progetto verte sulla realizzazione di un'infrastruttura software per la raccolta e la gestione di dati considerati sensibili e pertanto soggetti a rigorose normative in termini di privacy. Il sistema informativo realizzato è attualmente in uso presso le biblioteche della Provincia di Modena per la regolamentazione dell'accesso degli utenti ai servizi informatici.

La trattazione comprende la stesura dei requisiti tecnici e delle procedure, la descrizione del sistema di raccolta e manipolazione dei dati, del database e dell'interfaccia web di gestione.

Il progetto si completa con la descrizione di un sistema di ritenzione sicura dei dati a scopo di backup attualmente in uso presso le medesime infrastrutture.

Parole chiave:

Autenticazione

Dati personali

Privacy

Sicurezza

Backup

1. INTRODUZIONE 1
1.1. Scopo del progetto 1
1.2. Struttura del sistema informativo preesistente 3
1.3. Struttura del documento 5
1.3.1. Sistema di raccolta dati e autenticazione (progetto Bellerofonte) 6
1.3.2. Sistema di backup in sede remota (progetto Newbackup) 7
2. PROGETTO DEL SISTEMA INFORMATIVO 9
2.1. Politiche di gestione 9
2.1.1. Inserimento di utenti nella nuova anagrafica 10
2.1.2. Validazione dei dati degli utenti 13
2.1.3. Accesso autenticato a internet tramite i nuovi dati 16
2.1.4. Rettifica dei dati 17
2.2. Considerazioni tecnologiche 18
2.2.1. Autenticazione dell'accesso a internet 18
2.2.2. Database di utenti 19
2.2.3. Logica dell'applicativo di gestione 20
2.3. Topologia di riferimento 21
3. PROGETTO E REALIZZAZIONE DEL DATABASE 23
3.1. Introduzione ai servizi di directory 23
3.2. Definizione del servizio di directory 24
3.2.1. Schema OpenLDAP 24
3.3. Il servizio di directory del progetto Bellerofonte 29
3.3.1. Entità del database 29
3.3.2. Struttura del DIT 33
3.3.3. Classi 36
3.3.4. Attributi 46
3.4. Configurazione del servizio 49
3.4.1. Indici 49
3.4.2. Controllo di accesso 50
3.4.3. Replicazione 54
3.4.4. Altri parametri di configurazione 57

4. ESTRAZIONE DEI DATI DALLE ANAGRAFICHE PREESISTENTI 61

4.1. Architettura del sistema di trasferimento dati 61

4.1.1. Struttura del sistema di trasferimento 62

4.1.2. Tecnologie a supporto del Bus 64

4.2. Procedure 67

4.2.1. Importazione 67

4.2.2. Rilevamento dei duplicati 68

4.2.3. Scrittura sul DIT 72

5. PROGETTO E REALIZZAZIONE DELL'INTERFACCIA WEB 75

5.1. Requisiti 75

5.1.1. Funzionalità 75

5.1.2. Utenti e ruoli 76

5.1.3. Linee guida per la progettazione 78

5.2. Progetto 80

5.2.1. Struttura 80

5.2.2. Procedura di ricerca 84

5.2.3. Procedura di rilevamento duplicati 85

5.3. Architettura 89

5.3.1. Jakarta Tapestry 89

5.3.2. Componenti, classi, pagine e servizi 92

5.4. Realizzazione delle pagine 95

5.4.1. Specifica delle mappature in Tapestry 95

5.4.2. Composizione 97

5.4.3. Componenti per etichette e campi di input 99

5.4.4. Componenti per oggetti ciclici e opzionali 101

5.4.5. Radiobutton e checkbox 103

5.4.6. Componenti per la gestione del form 105

5.5. Formazione del personale 107

5.5.1. realizzazione di un manuale operativo 108

5.5.2. Corsi di formazione 109

6. ESTENSIONI AL PROGETTO BELLEROFONTE 113
6.1. Interfaccia amministrativa 113
6.1.1. Funzioni di ricerca 113
6.1.2. Funzioni batch 115
6.1.3. Funzioni di modifica e validazione 116
6.1.4. Funzioni di spostamento e ripristino 118
6.2. BiblioMedia 119
6.3. Servizi aggiuntivi per gli operatori delle biblioteche 122
6.3.1. Posta elettronica 122
6.3.2. Accesso nominativo alla catalogazione 124
7. BACKUP IN SEDE REMOTA 125
7.1. Processi di Backup 125
7.2. Caratteristiche di una strategia di Backup passivo 126
7.2.1. Backup incrementale o backup a snapshot 127
7.2.2. Strategie di replicazione 129
7.2.3. Sistema di copie storiche 131
7.2.4. Pull backup e push backup 132
7.2.5. Pianificazione del backup 133
7.2.6. Dimensionamento di un sistema di backup 134
7.3. Caratteristiche dell'infrastruttura da coprire 136
7.3.1. Esclusioni 136
7.3.2. Zone della rete 137
7.3.3. Sistemi operativi interessati 138
7.3.4. Natura dei dati 139
7.3.5. Vincoli alla pianificazione 140
7.4. Struttura del sistema di backup 141
7.4.1. Funzionalità 141
7.4.2. Moduli 143
7.5. Procedure 146
7.5.1. Backup 146
7.5.2. Backup storico 151

7.5.3. Verifica delle pool 153
7.5.4. Replicazione 154
7.5.5. Configurazione 156
7.6. Estensioni al sistema Newbackup 159
7.6.1. Gestione unificata dei log e dei messaggi di avviso 159
7.6.2. Gestione dei metadati 160
7.6.3. Servizio di backup alle biblioteche 161

1. INTRODUZIONE

1.1. Scopo del progetto

La rete bibliotecaria modenese, costituita nel 1989, comprende oltre 80 Biblioteche di diversa appartenenza istituzionale convenzionate con il Centro di Documentazione della Provincia di Modena. Comprende infatti la quasi totalità delle Biblioteche di Ente Locale (comprese quelle del Comune capoluogo), la Biblioteca Statale Estense, le Biblioteche dell'Università di Modena e Reggio Emilia , le Biblioteche scolastiche e Biblioteche di Istituzioni private come la Fondazione Collegio San Carlo, l'Accademia di Scienze Lettere ed Arti, la Fondazione Cassa di Risparmio di Modena, il Centro Documentazione Donna, ecc. Dal 2001, tramite apposita Convenzione con l'Istituto per i Beni Artistici, Culturali e Naturali della Regione Emilia-Romagna, la Biblioteca Estense Universitaria del Ministero per i Beni e le Attività Culturali, il Comune di Modena, la Fondazione Collegio San Carlo e il Centro Documentazione Donna di Modena, la rete bibliotecaria modenese si é ulteriormente qualificata e potenziata costituendosi in Polo Provinciale Modenese del Servizio Bibliotecario Nazionale.

La Rete Bibliotecaria modenese, che si configura come una Intranet, si avvale di un complesso di tecnologie informatiche, di rete e di programmi per la catalogazione del libro, la gestione del prestito automatizzato, la posta elettronica per gli operatori delle biblioteche, la consultazione in linea dei cataloghi bibliografici, l'accesso ad Internet da tutte le postazioni presenti presso ogni struttura.

L'infrastruttura di rete è attualmente caratterizzata da un'architettura eterogenea strutturata a stella, nella quale il CeDoc funge da centro di aggregazione dei servizi. L'architettura centralizzata favorisce la gestione dei sistemi informativi, accessibili localmente per manutenzione e aggiornamento.

Il progetto, nato con il nome DPS, ha lo scopo di adeguare l'infrastruttura di rete gestita dal CeDoc alle recenti normative in merito alla sicurezza, alla privacy, al trattamento dei dati personali e alla regolamentazione dell'accesso a internet.

Il CeDoc deve gestire ingenti informazioni anagrafiche relative lettori iscritti ai servizi telematici delle biblioteche per la cui gestione sono state formulate procedure in linea con il Decreto Legislativo 196/2003 (Documento Programmatico sulla Sicurezza). Il progetto DPS consiste della realizzazione di un sistema di aggregazione e raccolta di informazioni valide e del loro utilizzo per quanto prescritto nella Legge 155/2005 (Pacchetto Pisanu), in merito alla regolamentazione dell'accesso a internet.

Il progetto DPS ha condotto alla realizzazione di:

- Una struttura di raccolta per dati personali, costituita da interfacce con i sistemi informativi preesistenti;
- Una base dati di stoccaggio dei dati in formato aperto;
- Un sistema di validazione dei dati ad uso di personale autorizzato;
- Politiche di ritenzione dei dati in modo sicuro basate su una infrastruttura di "backup" locale con copia in sede remota.

L'obiettivo finale del progetto è di adeguare l'infrastruttura alla raccolta efficiente delle informazioni e al loro utilizzo per garantire credenziali valide per l'accesso a internet attraverso le postazioni messe a disposizione dalle biblioteche.

Il progetto a regime ha prodotto direttive procedurali da applicare per l'iscrizione degli utenti in biblioteca e per la gestione dei loro dati personali. Gli utenti che desiderano usufruire dei servizi di collegamento a internet presso la biblioteca possono farne richiesta e accedere alla procedura di abilitazione. Gli operatori effettueranno una validazione dei dati personali degli utenti e consegneranno loro un regolamento di utilizzo e una password segreta per l'accesso ai servizi.

La struttura di memorizzazione dei dati è allineata con quanto prescritto dalle normative sulle Pubbliche Amministrazioni Digitali ed è stata progettata in modo scalabile per garantire una base di partenza per utilizzi futuri delle informazioni e delle credenziali memorizzate.

1.2. Struttura del sistema informativo preesistente

L'infrastruttura di rete gestita dal CeDoc è costituita da una sede centrale che accentra l'accesso a tutti i servizi principali e da numerose strutture periferiche connesse alla struttura centrale attraverso reti wan. Le strutture periferiche non dispongono di repliche locali dei servizi e sono collegate con topologie ad albero in cui alcuni centri (di dimensioni maggiori) sono collegati direttamente con il CeDoc e condividono la banda con con centri più piccoli che si collegano a loro. Le strutture periferiche sono collocate presso le biblioteche comunali dell'intera provincia. Esse sono collegate alla struttura centrale attraverso varie modalità, tra cui collegamenti CDN, ISDN, ADSL e in fibra ottica. Benché un progetto di unificazione e standardizzazione delle strutture di rete sia attualmente in atto, l'attuale panorama consente alle biblioteche con linee di collegamento meno performanti l'utilizzo dei cinque servizi principali attualmente in uso:

- Gestione informatizzata del prestito dei libri;
- Gestione informatizzata della catalogazione dei libri;
- Accesso a internet;
- Posta elettronica;
- Banche dati ad accesso riservato.

I quattro servizi sono considerati la base accessibile a tutte le biblioteche si completano con molteplici opzioni accessorie alla gestione dei sistemi informativi e con il supporto tecnico hardware e software, tuttavia la descrizione di questi servizi non è parte del documento.

La gestione del prestito libri è attualmente effettuata attraverso due distinti applicativi:

- Sebina. Strumento di collegamento al sistema bibliotecario nazionale, in uso per la catalogazione delle opere in tutta la provincia e per il prestito nelle sole biblioteche della zona del Comune di Modena. Le componenti software per la gestione di questo sistema sono proprietarie e gestite dalla ditta Data Management;
- Auriga. Strumento di prestito in uso a tutte le biblioteche escluse quelle dell'area del Comune di Modena. Le componenti software per la gestione di questo sistema sono state realizzate e sono mantenute internamente in collaborazione con le ditte Team Software srl e Datacode srl.

Entrambi i sistemi dispongono di una anagrafica utenti, le cui informazioni sono divenute oggetto di procedure di salvataggio e backup nel corso del progetto. Questi due sistemi sono di uso corrente, eppure nel corso del progetto sono stati considerati sistemi legacy dai quali gestire procedure di importazione dei dati. In merito all'interazione con questi sistemi, l'obiettivo è quello di realizzare una unica anagrafica ottenuta dall'unione delle due preesistenti, depurata da duplicati e costituita soltanto da informazioni valide e consistenti con i documenti di identità.

Le due anagrafiche in uso costituiscono una buona base di partenza per l'ottenimento dei dati, tale da rendere ingiustificato l'onere di raccoglierli nuovamente per popolare il nuovo database. Inoltre l'obbligo di legge di legare l'utilizzo dei servizi all'iscrizione in biblioteca giustifica una relazione stretta tra anagrafiche del prestito e anagrafica dei servizi. Queste due considerazioni sono state tradotte nel vincolo progettuale di non rendere possibile l'inserimento libero di entità nella nuova anagrafica consentendo solo l'importazione da anagrafiche esistenti. La scelta ha reso necessario lo sviluppo di un sistema in grado di integrare i dati delle due anagrafiche con rilevamento duplicati tra le due e interni alla stessa anagrafica. Si noti che il vincolo riguardante l'iscrizione al prestito si riferisce a una qualunque biblioteca che utilizzi il sistema, mentre il diritto di accesso ai servizi derivante è riferito a tutte le biblioteche. In altre parole un utente che desidera accedere ai servizi e non è iscritto in biblioteca deve affrontare una sola procedura di iscrizione e abilitazione in una biblioteca e potrà accedere ai servizi in tutte.

Nella completa integrazione con il sistema informativo esistente, il progetto DPS esprime i seguenti vincoli:

- Gestire l'integrazione con due anagrafiche facenti capo a sistemi legacy di prestito e catalogazione;
- Consentire la memorizzazione di nuove entità (utenti) soltanto se riferite a persone iscritte al prestito presso almeno una biblioteca;
- Consentire l'utilizzo delle informazioni memorizzate al fine di offrire all'utenza registrata l'accesso ai servizi regolamentato in termini di legge;
- Consentire la gestione dei dati da parte di personale abilitato, la validazione, la rettifica e la riconferma nei termini di legge;
- Favorire l'utenza occasionale dei servizi, semplificando le procedure di gestione delle informazioni (come il cambio password);
- Disporre di un sistema di memorizzazione dei dati personali in un formato aperto, come sancito dalle normative vigenti a regolamentare tale procedura per le Pubbliche Amministrazioni;
- Disporre di una metodologia di "backup" o salvataggio dei dati sensibili protetta e replicata in sede remota;
- Mantenere informazioni adatte per l'estensione futura del parco servizi.

1.3. Struttura del documento

Il progetto DPS ha dato vita a due prodotti software fondamentali, ciascuno con uno scopo specifico. La trattazione riguardante progettazione e sviluppo è pertanto separata per prodotto. La trattazione riguardante le politiche formulate per la loro gestione è altrettanto separata. Le due componenti sono:

- Sistema di raccolta dati e autenticazione dell'accesso a internet;
- Sistema di ritenzione dati e backup in sede remota.

Segue una breve introduzione di ciascuna delle due componenti.

1.3.1. Sistema di raccolta dati e autenticazione (progetto Bellerofonte)

Il progetto Bellerofonte ha assorbito la maggior parte degli sforzi profusi nell'intero processo di adeguamento. Il progetto ha portato alla realizzazione di un prodotto software integrato in grado di gestire l'intera procedura di importazione dei dati dai sistemi legacy, gestione e validazione dei dati, impiego dei dati per l'autenticazione dell'unico servizio di partenza: l'accesso a internet.

Il prodotto software risultante è una componente software complessa costituita da:

- Un'interfaccia con i sistemi legacy limitata alle anagrafiche;
- Una nuova e scalabile struttura di stoccaggio dei dati;
- Una tecnologia di trasferimento dati dall'interfaccia alla nuova struttura;
- Un'interfaccia utente e logica applicativa per gestire i dati presso la nuova struttura;
- Un modello di autenticazione per l'accesso a internet in grado di interagire con al nuova struttura.

La combinazione di queste componenti costituisce il risultato tecnico del progetto. Questo documento riporta informazioni di carattere progettuale in merito alla formulazione delle specifiche (capitolo 2), il progetto e la realizzazione del database (capitolo 3), i metodi e gli strumenti di estrazione dei dati (capitolo 4), il progetto e la realizzazione dell'interfaccia web (capitolo 5) e si conclude con una breve introduzione a progetti futuri o attualmente in elaborazione che si appoggiano sul sistema informativo realizzato (capitolo 6).

Il progetto Bellerofonte è stato realizzato in parte all'interno del CeDoc in parte con la collaborazione della ditta Datacode srl. Nella fattispecie:

- Il progetto del sistema è stato realizzato internamente: l'autore ha svolto un ruolo determinante nella formulazione e nella documentazione delle politiche di gestione;
- Il progetto e la realizzazione del database sono interamente a cura dell'autore;

- L’autore ha diretto la stesura delle procedure attuate nel sistema di trasferimento dati, la cui realizzazione è stata delegata alla ditta esterna;
- L’autore ha progettato e realizzato le pagine dell’interfaccia web e ha diretto al definizione della logica applicativa, la cui realizzazione è stata delegata alla ditta esterna;
- L’autore ha curato le fasi di formazione del personale, prima nella redazione del manuale operativo, poi nello svolgimento in prima persona dei corsi di formazione.

1.3.2. Sistema di backup in sede remota (progetto Newbackup)

Il progetto Newbackup ha portato alla realizzazione di un sistema di memorizzazione dei dati rilevanti estensibile a tutti i servizi gestiti presso il CeDoc. Il prodotto software è un insieme di procedure automatizzate per gestire il prelievo di qualsiasi genere di informazione delle apparecchiature che gestiscono i servizi interni all’infrastruttura di rete.

I dati vengono prelevati attraverso un canale cifrato (ove disponibile) e vengono memorizzati in un dispositivo adatto alla ritenzione. Il software gestisce gli errori di trasferimento attraverso notifiche ed è in grado di riprodurre repliche parziali o totali dei dati memorizzate in versioni identiche del software in esecuzione presso sedi remote.

Le politiche di memorizzazione dei dati formulate per il sistema sono applicate ai dati personali gestiti nell’infrastruttura, a documenti gestionali e amministrativi e a materiale tecnico per la manutenzione delle infrastrutture harware/software.

La progettazione e la realizzazione dell’infrastruttura di backup è stata curata dall’autore, internamente al CeDoc. La manutenzione e la realizzazione di nuove componenti sono altresì in gestione all’autore.

2. PROGETTO DEL SISTEMA INFORMATIVO

2.1. Politiche di gestione

Il primo fondamentale passaggio nella progettazione del nuovo sistema è stato lo studio dell'impatto delle nuove politiche di gestione sull'infrastruttura esistente. E' opportuno notare che l'aggiunta di un sistema di raccolta dati, per quanto operante in modo automatizzato, comporta un disagio all'utenza che ha il compito di effettuare la raccolta. Il disagio, se mal gestito, può risultare in un atteggiamento refrattario e a un difficile rapporto con lo strumento lavorativo, con le complessità di gestione derivanti.

Il procedimento di studio delle politiche di gestione si compone di due fasi:

- Analisi delle procedure dal punto di vista tecnico;
- Analisi dell'interazione richiesta agli utenti per le procedure e conseguente semplificazione strutturale delle stesse.

Il sistema esistente è caratterizzato dalla presenza preponderante di alcune metodologie assodate, le quali non richiedono addestramento del personale e incontrano già l'approvazione degli utenti che le devono attuare. Per questo motivo si è scelto di utilizzare in più possibile gli strumenti di raccolta dati già esistenti, sebbene diversi tra loro. La scelta è giustificata dalla familiarità degli utenti con i due sistemi e dalla regolarità che una procedura di "pulizia" dei dati può portare in un database esistente. In altre parole, rendendo i dati delle anagrafiche utilizzabili per il nuovo sistema, gli operatori hanno anche provveduto a migliorare la qualità dei dati già esistenti, sottoponendoli per obbligo a un processo di validazione.

Le procedure più rilevanti sono:

- Inserimento di utenti nella nuova anagrafica;

- Validazione dei dati degli utenti;
- Accesso autenticato a internet tramite i nuovi dati;
- Rettifica dei dati.

2.1.1. Inserimento di utenti nella nuova anagrafica

Agli operatori si richiede di inserire gli utenti nella nuova anagrafica, gli utenti non possono accedere all'inserimento in modo autonomo. Dal punto di vista tecnico si tratta di importare i dati certamente validi da una delle anagrafiche preesistenti presso la nuova. Questa importazione è scatenata da un evento relativo al singolo record: ogni profilo anagrafico viene candidato per l'importazione ogni volta che viene modificato. All'avvenire di questo evento, circoscritto all'anagrafica delle persone singole, il sistema è in grado di identificare il dato (o gruppo di dati, l'intera anagrafica di una persona) come candidato per l'importazione. A questo punto interviene una procedura di verifica per l'esistenza di tutti i dati necessari, ovvero:

- Cognome;
- Nome;
- Data di Nascita;
- Luogo di nascita;
- Sesso;
- Tipo di documento;
- Numero del documento;
- Indirizzo.

Il concetto di "dati necessari" è relativo alla totalità di dati reperibili presso un documento di identità valido. Per la prima versione del sistema si è scelto di considerare validi soltanto la Carta di Identità, il Passaporto e la Patente. Questi dati devono essere presenti nell'anagrafica di partenza per consentire l'importazione in quanto si desidera che i dati importati siano completi ai termini di legge. E' stata fatta un'eccezione per le anagrafiche di Sebina in merito al Luogo di Nascita: campo obbligatorio ma non presente nel database di partenza. Il campo deve essere completato in seguito.

Si noti che nelle anagrafiche preesistenti esiste una solo campo per l'indirizzo e uno per la città. Questi due dati devono essere interpretati come "Indirizzo e città di domicilio". La loro presenza nelle anagrafiche degli utenti iscritti alle biblioteche è infatti necessaria per la spedizione di solleciti alla restituzione dei volumi, pertanto possono non corrispondere al reale dato di residenza riportato sul documento. Nell'ipotesi di migliorare la qualità dei dati nelle anagrafiche preesistenti deve essere tenuta in considerazione la necessità di operare a supporto della biblioteca, pertanto le politiche formulate non possono in alcun modo imporre che venga memorizzato un dato non di valore a discapito di uno fondamentale. Per garantire l'integrità ai termini di legge si è deciso di delegare alle interfacce utente del nuovo sistema questo processo di inserimento.

Nelle procedure di inserimento si richiede agli operatori di richiamare il profilo dell'utente (o iscriverlo se non esiste), verificare i dati e salvare, indipendentemente dalla presenza di modifiche sostanziali all'anagrafica. All'atto del salvataggio interviene la procedura di trasferimento, configurata come segue:

- Per Auriga: generazione a tempo di esecuzione di un file descrittore della entry contenente una riga di testo a campi separati, ciascuno corrispondente a uno degli attributi trasmessi dall'anagrafica precedente;
- Per Sebina: generazione periodica di un file contenente una riga simile a quella di Auriga per ciascuno dei profili modificato nell'arco della giornata (un profilo per cui viene eseguito il salvataggio compare in tutti i file generati fino a fine giornata).

I file descrittori vengono assorbiti dal sistema che li analizza prelevando i parametri necessari per creare il profilo nella nuova anagrafica, verificando:

- Che il nome e il cognome siano nella forma "Cognome, Nome", vincolo necessario alla corretta distinzione delle due parti del nome proprio;
- Che tutti gli altri dati esistano (con l'eventuale eccezione del Luogo di Nascita).

Se il confronto ha esito positivo, il sistema procede con la realizzazione di un nuovo profilo e con le necessarie procedure di verifica dei duplicati che saranno descritte in seguito.

L'inserimento e la rettifica dei dati sono delegati alle anagrafiche preesistenti, pertanto gli operatori intervengono su di esse per la prima fase della verifica. Il sistema reagisce alle aggiunte e alle modifiche in modo automatico.

L'intera procedura richiesta, per quel che riguarda l'inserimento è la seguente:

- L'utente fa richiesta di navigare in internet, il diritto di navigare lo rende soggetto alla raccolta dati utilizzando il nuovo sistema. Al termine della procedura di inserimento l'utente potrà essere abilitato alla navigazione (si vedano i paragrafi successivi);
- L'operatore deve verificare lo stato di iscrizione al prestito dell'utente. In condizioni ideali si richiede di richiamare l'anagrafica preesistente se disponibile. Nella realtà la maggior parte degli operatori ha difficoltà a richiamare anagrafiche di utenti scritti presso biblioteche differenti, pertanto tende a reiscrivere l'utente nella propria biblioteca. Benché non gli sia concesso di modificare i dati sulle anagrafiche preesistenti, il sistema è in grado di reagire a questa eventualità;
- Sia in caso di nuova iscrizione che di richiesta dei dati, l'operatore deve verificare che essi siano formulati nel modo richiesto dalla nuova anagrafica e che corrispondano ai dati riportati sul documento fornito dall'utente, poi salvare le modifiche;
- Il sistema reagisce al salvataggio e attiva la procedura di inserimento.

Agli operatori è pertanto richiesta soltanto una verifica dei dati prima di inviarli al nuovo sistema. Se è necessario iscrivere nuovi utenti, la procedura non differisce da quella già presente. Gli operatori devono in ogni caso riportare i dati come presenti sul documento, ad eccezione dell'indirizzo di domicilio, che sarà trattato in seguito.

La procedura di iscrizione di utenti minorenni differisce lievemente. Per gli utenti minorenni non è richiesto un documento di identità, tuttavia se esso è noto può essere memorizzato. La responsabilità per l'operato di utenti minorenni ricade sempre sul genitore. La traccia necessaria a memorizzare questo collegamento è segnalata presso la procedura di validazione.

In questa frazione della procedura globale gli utenti finali non hanno ruolo, se non nel richiederne l'inizio. Agli utenti può essere consegnato il regolamento per l'accesso a internet, affinché possano prenderne visione.

2.1.2. Validazione dei dati degli utenti

Per completare l'iscrizione gli operatori devono attivare il profilo degli utenti presso la nuova anagrafica. L'operazione è resa necessaria dalla natura semi-automatica del processo di importazione. l'obiettivo è accertarsi che soltanto gli utenti che hanno fornito un documento possano accedere ai servizi soggetti ad autenticazione. Per completare l'attivazione agli operatori è richiesto di:

- Accedere allo strumento messo a disposizione per lo scopo, l'accesso è consentito soltanto agli operatori;
- Richiamare l'anagrafica dell'utente precedentemente importata;
- Verificare che i dati siano stati importati correttamente;
- Completare i dati mancanti;
- Affidare una password all'utente;
- Abilitare l'utente per l'accesso a internet;
- Salvare le modifiche.

Questi passaggi, sebbene di apparente complessità, sono attuabili compilando il modulo messo a disposizione degli operatori. La procedura di attivazione si traduce quindi in:

- Richiamare l'anagrafica dell'utente;
- Completare i campi vuoti;
- Salvare le modifiche.

Nella fattispecie, la procedura è stata formalizzata come segue:

Un operatore per qualificarsi come tale deve fornire un nome utente e una password. Per coerenza questi dati sono identici a quelli che la stessa persona utilizza per gli altri servizi,

pertanto lo stato di operatore si ottiene nella forma di un'abilitazione alla gestione dei servizi applicata ad un account esistente.

La ricerca delle anagrafiche deve consentire l'inserimento di dati di facile reperibilità. Gli operatori possono cercare per Cognome e Nome oppure per tessera bibliotecaria, esattamente come avviene negli strumenti che sono già abituati ad utilizzare.

Il risultato della ricerca deve consentire la rapida distinzione dell'utente di cui modificare i dati, con particolare attenzione alla presenza di duplicati e alla rilevazione di rettifiche. Il sistema deve richiedere il minimo input all'utente per risolvere qualsiasi caso di ambiguità.

L'anagrafica dell'utente deve essere comprensiva di ogni dato conosciuto per l'utente, ciò significa che se si dispone di informazioni non obbligatorie al funzionamento, tali informazioni vengono mostrate ugualmente (ad esempio il tipo e numero di documento, se esistono per un minorenne). I campi obbligatori per cui non è presente un valore devono essere compilati dall'operatore (ad esempio il luogo di nascita per anagrafiche importate da Sebina). L'indirizzo ottenuto dalle anagrafiche precedenti è considerato un indirizzo di domicilio, pertanto si richiede che venga esplicitato un indirizzo di residenza. La specifica deve favorire i casi in cui indirizzo di domicilio e indirizzo di residenza coincidono.

Il sistema deve essere in grado di generare un nome (uid) che gli utenti potranno utilizzare per collegarsi a internet, associato alla password. Questo nome utente deve essere generato a partire dal nome completo (Nome e Cognome) della persona attraverso le seguenti regole, applicate in sequenza:

1. Le lettere accentate vengono sostituite dalle corrispondenti senza accento;
2. Le parole separate da apostrofi sono unite e altri caratteri non alfabetici sono rimossi;
3. Nomi completi costituiti da un sola parola per il nome e una sola per il cognome generano uid nella forma "cognome.nome";
4. Nomi completi costituiti da più parole per il nome generano uid nella forma "cognome.nome composto" dove nome composto e' costituito dalla prima parola del nome e dalla prima lettera delle successive;

5. Nomi completi costituiti da più parole per il cognome generano uid nella forma "cognome composto.nome” dove cognome composto e' costituito dalla prima lettera di tutte le parole del cognome tranne l'ultima, riportata interamente.
6. Se le parole da abbreviare sono composte da meno di tre lettere non vengono abbreviate;
7. Nomi completi costituiti da più parole per il nome e più parole per il cognome generano uid nella forma "cognome multiplo.nome multiplo", con le abbreviazioni del caso;
8. L'uid non può mai essere cambiato.

L’uid non deve essere mai duplicato: il sistema verifica l’esistenza dell’uid generato tra tutti gli utenti presenti e aggiunge un numero di serie in caso il dato esista già.

L’affidamento di una password deve essere personale e segreto. Gli operatori non devono conoscere la password che viene affidata agli utenti. La password viene pertanto affidata in busta chiusa: agli operatori vengono fornite buste numerate, ciascuna contenente una password e alcune informazioni per l’utilizzo. Gli operatori devono soltanto riportare il numero scritto esternamente alla busta presso il modulo di raccolta per affidare la password contenuta in modo automatico all’utente. Gli utenti apriranno la busta privatamente e dovranno mantenere la password segreta. Agli utenti deve essere consentito modificare la propria password, in base alle seguenti regole:

1. La password deve essere lunga almeno otto caratteri;
2. La password non può essere uguale all’uid;
3. Ogni nuova password deve essere diversa dalla precedente;
4. La password deve contenere almeno un numero;
5. La password deve contenere almeno una lettera maiuscola e una minuscola

La procedura di cambio password non fa parte delle competenze degli operatori e può essere svolta autonomamente dagli utenti.

Infine, l’abilitazione per l’accesso ad internet deve essere immediata e la stessa interfaccia deve consentire l’abilitazione come operatori, ovvero alla gestione del sistema. In tal

modo gli operatori possono abilitare e disabilitare i colleghi. L'abilitazione per l'accesso a internet deve disporre delle tre opzioni:

1. Abilitato: l'utente può navigare in internet utilizzando i propri dati di autenticazione;
2. Disabilitato: il profilo dell'utente non è utilizzabile per l'accesso a internet;
3. Sospeso (fino a una data specifica): fino alla data specificata l'utente non potrà navigare in internet, al raggiungimento della data la sospensione sarà rimossa automaticamente.

Al momento della progettazione l'unico servizio soggetto ad autenticazione era l'accesso a internet, pertanto la maggior parte delle procedure sono state attivate per gli operatori con una simbologia e nomenclatura relative a tale servizio. Tutte le procedure e le interfacce hanno tuttavia il requisito fondamentale di scalabilità rispetto al numero di servizi gestiti.

Il salvataggio dei dati, e la relativa conferma del sistema determinano il completamento della procedura di attivazione. Agli operatori deve essere notificato che ogni modifica e conferma attuata nell'ambito del nuovo sistema è legata all'operatore che la svolge e che la procedura informatica impone in ogni caso la procedura amministrativa della verifica dei dati del documento (quando disponibile) e firma del regolamento per l'accesso ai servizi.

2.1.3. Accesso autenticato a internet tramite i nuovi dati

Al termine della procedura l'utente può iniziare subito la navigazione in internet (o può effettuare subito l'accesso a qualsiasi servizio del CeDoc si avvalga del sistema di autenticazione). Gli utenti vengono esortati a cambiare immediatamente la propria password, tuttavia le credenziali di cui dispongono hanno valore immediato.

Per la prima navigazione l'utente deve aprire la busta sigillata contenente la password e fornirla all'avvio del programma di navigazione, corredata al nome utente che gli è stato comunicato. Le credenziali di accesso non devono essere verificate nuovamente dal programma fino al termine dell'utilizzo.

Dal momento in cui viene iscritto l'utente ha due scadenze temporali da rispettare:

- Obbligo di cambio password. L'utente deve cambiare la propria password al massimo ogni sei mesi dall'ultimo cambio. Il cambio password è soggetto alle regole espresse in precedenza;
- Obbligo di verifica dei dati. Ogni due anni gli utenti devono presentarsi ad un operatore muniti di documento per la verifica dei propri dati personali. L'operatore dovrà accedere al profilo dell'utente e verificare che i dati aderiscano con quelli presenti sul documento e in caso contrario operare una rettifica. Si considera operazione di conferma dei dati ogni salvataggio eseguito da un operatore.

Se un utente non rispetta uno qualunque dei due obblighi, l'accesso ai servizi viene interdetto finché l'obbligo non viene rispettato.

2.1.4. Rettifica dei dati

La procedura di rettifica parte sempre dalle anagrafiche preesistenti. Per modificare un dato qualsiasi di quelli noti per gli utenti, gli operatori devono accedere alle anagrafiche disponibili e operare le modifiche. Al momento del salvataggio le modifiche vengono replicate sulla nuova anagrafica e all'utente viene temporaneamente interdetto l'accesso ai servizi.

La procedura mira ad evitare che gli utenti possano usufruire dei servizi senza essere coperti da credenziali valide e confermate sulla responsabilità di un operatore. In caso di modifiche di lieve entità (come l'indirizzo di domicilio) la sospensione non viene applicata.

In caso vengano effettuate rettifiche di errori, ad esempio la modifica di un nome, il sistema affronta le entry generate come possibili duplicati e richiede una scelta agli operatori per eliminare l'ambiguità in modo semi-automatico. La rimozione dell'ambiguità consiste nella decisione su quale entità considerare valida per futuri utilizzi dei servizi. Le entità scartate da questo processo decisionale non devono essere rimosse per favorire l'analisi della storia dei profili utente nel sistema.

2.2. Considerazioni tecnologiche

La realizzazione del sistema descritto richiede l'analisi e lo sviluppo di alcune componenti fondamentali ed è stato di primaria importanza sviluppare ciascuna di esse sulla base di prodotti affermati e standard condivisi. Questo paragrafo riporta alcune considerazioni che specificano le scelte fatte in termini di tecnologie adottate.

2.2.1. Autenticazione dell'accesso a internet

L'autenticazione dell'accesso a internet condiviso da più postazioni distribuite su rete geografica può essere garantita soltanto se viene abilitato un sistema di controllo in prossimità dell'accesso alla connessione condivisa. L'infrastruttura esistente ha consentito questa scelta, caratterizzata da:

- Un modulo di verifica delle credenziali di accesso per i principali protocolli di comunicazione in rete;
- Un modulo di logging o mantenimento di dati a norma di legge sull'uso fatto di tali credenziali.

Tali funzioni sono eseguite egregiamente dal prodotto open source Squid Cache[i]. Il prodotto è essenzialmente un server proxy, nato per migliorare le prestazioni di un collegamento a internet mantenendo in prossimità della rete locale di una organizzazione una copia dei siti web visitati più di frequente. Il miglioramento di prestazioni è relativo alle richieste fatte per tali siti.

Trattandosi di uno strumento collocato tra tutti i richiedenti una connessione e l'unica via disponibile e condivisa di collegamento, un proxy può eseguire le funzioni di filtraggio e autenticazione richieste dal sistema.

L'obiettivo è forzare gli utenti a utilizzarlo come unico passaggio e configurarlo per:

[i] Squid Cache è uno dei più noti software per il caching locale. Sorgenti, binari e documentazione disponibili presso http://www.squid-cache.org/

- Garantire l'accesso solo agli utenti in grado di fornire un uid (nome utente) e una password validi;
- Memorizzate qualsiasi richiesta e risposta che lo attraversano utilizzando i protocolli più comuni (quali http, ftp e altri).

Squid cache è inoltre in grado di applicare metodi di filtraggio dei contenuti visualizzati, decidendo di interdire la visualizzazione di siti che contengono materiale la cui fruizione non è consentita. E' sufficiente aggiungere un modulo di filtraggio[ii] per abilitare questa funzionalità e disporre del relativo sistema di logging.

2.2.2. Database di utenti

I dati personali degli utenti devono essere mantenuti in almeno un formato aperto, cioè accessibile attraverso tecnologie basate su open standard. Il database delle anagrafiche è essenzialmente una rubrica di grandi proporzioni, ad eccezione della componente in grado di memorizzare password e altri dati a supporto alla logica applicativa. In generale è possibile dipingerne la struttura come un albero con molte foglie, ciascuna che rappresenta l'entità associata a un utente. Ogni entità ha numerosi attributi che definiscono i dati mantenuti dal sistema.

Questa struttura ad albero è facilmente espressa nei termini di uno spazio nomi OSI X.500, costituito da percorsi univoci che definiscono entry isolate, ciascuna delle quali è specificata da alcuni attributi. Una implementazione open source di questo spazio nomi è OpenLDAP[iii]. Questo prodotto consente di realizzare, mantenere e interrogare un servizio di directory in cui numerose entry sono identificate con sintassi e semantica X.500[iv] e memorizzate su un backend (tipicamente Berkeley DB[v]). Per servizio di directory si intende un databa-

---

[ii] Per questo ipo di funzionalità sono disponibili moduli aggiuntivi per Squid. Tra questi è presente SquidGuard. Sorgenti, binari e documentazione disponibili presso http://www.squidguard.org/

[iii] OpenLDAP è la più nota ed efficiente implementazione Open Source del protocollo OSI x.500. Il prodotto comprende un directory server LDAP, librerie per la realizzazione di client e client a linea di comando. Sorgenti e documentazione disponibili presso: http://www.openldap.org/

[iv] Maggiori informazioni sul protocollo OSI X.500 disponibili presso http://sec.cs.kent.ac.uk/x500book/

[v] Berkeley DB è un motore per la realizzazione di database completamente Open Source. Il prodotto è mantenuto da SleepyCat Software. Maggiori informazioni presso http://www.sleepycat.com/products/bdb.html

se ottimizzato per la lettura e la ricerca, adatto per memorizzare grandi quantità di dati senza forti relazioni tra entry.

Per operare sulle entry, OpenLDAP utilizza l'implementazione del protocollo LDAP (Lightweight Directory Access Protocol). Questo protocollo è ampiamente supportato da prodotti di middleware, in numerosi linguaggi di programmazione.

OpenLDAP dispone inoltre di un sistema di replicazione e sincronizzazione avanzato e preciso, ulteriore motivo per cui si è scelto di operare utilizzando questa tecnologia.

2.2.3. Logica dell'applicativo di gestione

Per consentire l'interoperabilità di OpenLDAP e dei sistemi legacy e per realizzare la logica applicativa per la gestione da parte degli operatori è stato necessario scegliere una tecnologia flessibile. La tecnologia più affermata e con il maggior campo di applicabilità su middleware enterprise è J2EE (Java 2 Enterprise Edition).

Questa tecnologia ha consentito la realizzazione degli strumenti per la comunicazione dei nuovi servizi ed è stata la base per lo sviluppo di un'interfaccia web per la gestione da parte degli operatori.

La maggior parte dello sviluppo nella piattaforma J2EE è stata volta in collaborazione con la ditta Datacode s.r.l. Il codice sorgente prodotto nell'ambito di questa collaborazione non può essere divulgato integralmente ma nelle sezioni successive si farà riferimento alla struttura globale dell'applicativo.

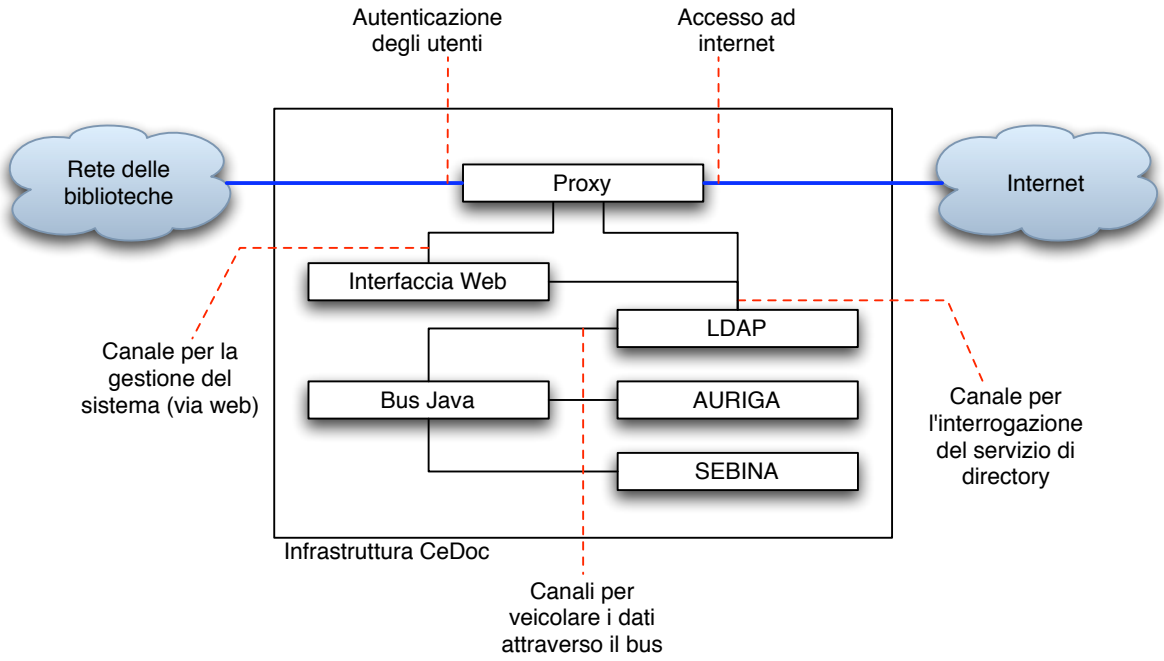
In particolare si farà riferimento al "Bus Java" per intendere la tecnologia di trasferimento informazioni da e per il nuovo sistema informativo e all'"Interfaccia Web" per intendere la logica applicativa e di presentazione dell'interfaccia per gli operatori. Quest'ultima è stata realizzata utilizzando la tecnologia Jakarta Tapestry, prodotto per il rendering Web della fondazione Apache e interamente scritto in Java.

2.3. Topologia di riferimento

La formalizzazione delle procedure prescritte in un progetto per il sistema informatico ha prodotto l'isolamento delle seguenti componenti:

- Sistema di autenticazione e filtraggio dei contenuti (proxy);
- Sistema per la gestione dei dati (Interfaccia Web);
- Sistema per lo stoccaggio dei dati (LDAP);
- Sistema per veicolare i dati dalle basi legacy (Bus Java).

Figura 2.1: Topologia del sistema software per il progetto Bellerofonte



Il canale blu indica il percorso effettuato dalle connessioni degli utenti nelle biblioteche per l'accesso a internet. Agli utenti normali deve essere trasparente l'esistenza dell'infrastruttura di gestione al disotto della componente Proxy.

Gli operatori raggiungono anche il livello di Interfaccia Web per la gestione dei dati. Essi percepiscono l'interazione con questa componente come l'accesso a un sito web, tale ac-

cesso è veicolato attraverso il proxy come tutti gli altri accessi web, ma non richiede autenticazione, trattandosi di un sito web “interno”.

Le restanti componenti sono totalmente trasparenti a ogni categoria di utenti e operano per la gestione del sistema. In particolare, il Bus Java opera il trasferimento e la conversione tra i due sistemi Auriga e Sebina e il nuovo LDAP, limitatamente ai dati anagrafici.

In questa topografia funzionale sono escluse informazioni di implementazione, come presenza di Firewall, posizione fisica sulle macchine dei servizi elencati.

3. PROGETTO E REALIZZAZIONE DEL DATABASE

3.1. Introduzione ai servizi di directory

Un servizio di directory è un database progettato per favorire le procedure di interrogazione e ricerca. Tipicamente la sua struttura si adatta a realtà in cui le modifiche sono aggiornamenti di intere entità e sono piuttosto rare, se consentite. L'efficienza nella ricezione dei cambiamenti è sacrificata a favore delle prestazioni in ricerca.

Il prodotto software OpenLDAP implementa lo spazio nomi OSI X.500 per realizzare un servizio di directory nel quale l'elemento base è definito entry. Ogni entry è identificata da un percorso univoco nello spazio nomi, detto Distinguished Name (DN). In questo documento si farà riferimento alle entry del database anche con l'appellativo DN. Le entry sono specificate da attributi, ciascuno dei quali ha un tipo (tipo di dato) e uno o più valori.

Le entry sono organizzate in uno spazio gerarchico strutturato ad albero, il Directory Information Tree (DIT). La radice dell'albero è l'organizzazione di riferimento, gli ulteriori livelli costituiscono specializzazioni. E' possibile affidare vari significati alle diramazioni dello spazio nomi, per questo la sua progettazione è un passaggio importante della realizzazione della base dati.

La definizione degli attributi per i quali ogni entry può assumere valori è delegata alla sua specifica di appartenenza a categorie dette objectClass. Le objectClass definiscono un insieme di attributi per i quali deve essere definito un valore obbligatoriamente e un insieme di attributi di corredo. Per le objectClass esiste l'ereditarietà in termini di attributi e una entry può appartenere a più di una objectClass, di cui almeno una deve essere strutturale, o dichiarativa della struttura della entry. Le objectClass e gli attributi che determinano devono essere

definite in elementi detti schema. La scelta di attributi da schema esistenti o la definizione di nuovi schema è il secondo punto fondamentale nella definizione di un servizio di directory.

Lo strumento OpenLDAP implementa la gerarchia di X.500 nella definizione di un albero che può essere interrogato noti due concetti: search base e search filter. Il primo identifica il sottoalbero da interrogare in una query di ricerca, il secondo definisce il filtro in base al quale le entry devono essere selezionate. La differenziazione funzionale tra le entry spesso viene espressa come appartenenza a diverse search base. Le caratteristiche del filtro sono spesso utilizzate per identificare le entry da cercare, procedimento molto utile per sistemi di autenticazione basati su LDAP.

Lo stesso approccio della ricerca è utilizzato come punto di partenza per modifiche, cancellazioni e inserimenti. Ogni interazione con un sistema OpenLDAP rispetta il modello client server in cui un sistema client dispone delle primitive di interrogazione e un sistema server rimane in attesa di richieste. Un sistema server può essere client di altri sistemi server; è il caso della funzionalità syncrepl utilizzata per fornire un servizio di replicazione del database. La replicazione può essere utilizzata per generare copie di un database a scopo di backup o per soddisfare interrogazioni e favorire il bilanciamento di carico.

3.2. Definizione del servizio di directory

La realizzazione di un servizio di directory richiede la definizione di alcuni parametri fondamentali, alcuni dei quali competono alla configurazione dello strumento utilizzato, altri alla specifica del servizio di directory. In questa sezione sono riportate le nozioni necessarie alla progettazione di un servizio di directory, quella successiva tratterà della configurazione della directory realizzata. Le nozioni di base sono completate dalle scelte intraprese nel corso del progetto.

3.2.1. Schema OpenLDAP

Qualunque distribuzione di OpenLDAP contiene alcuni schema predefiniti asserviti alle comuni funzionalità di un servizio di directory. Attraverso tali schema è possibile realizzare un DIT contenente entità formulate in modo da interagire con i più comuni prodotti nei campi

dell'autenticazione e della gestione rubriche. E' consigliabile utilizzare quanto più possibile gli attributi formulati negli schema già presenti, allo scopo di ottenere un database scalabile e capace di interrogare con sforzi molto ridotti con una grande varietà di software. L'esempio più frequente di questa compatibilità è dato dal largo utilizzo che si fa degli attributi uid e userPassword, fondamentali per definire credenziali di accesso per qualsiasi sistema affermato che utilizzi procedure di autorizzazione e autenticazione basate su DLAP.

Tabella 3.1: Schema predefiniti in OpenLDAP

Nome del file	***Descrizione***
core.schema	Componente principale di OpenLDAP
cosine.schema	Schema Cosine e X.500 per Internet
inetorgperson.schema	Utile schema per la definizione di rubriche
misc.schema	Attributi a supporto, misti
nis.schema	Schema per Network Information Services
openldap.schema	Schema sperimentale di OpenLDAP

Tra gli schema predefiniti, i più utili nella realizzazione di un database risultano essere nis.schema e inetorgperson.schema. Tutti gli schema dovrebbero essere ugualmente inclusi all'interno di ogni comune configurazione, a causa delle forti dipendenze interne tra uno schema e l'altro.

Alcuni ambiti organizzativi possono richiedere di formulare una raccolta di specifiche personalizzata, la quale può essere compilata in due modi non necessariamente disgiunti: eredità da attributi esistenti e realizzazione di nuovi attributi. Nel primo caso si tratta di sfruttare il concetto di ereditarietà per utilizzare attributi già presenti in altri schema come base per realizzarne altri. Nel secondo caso si tratta di definire nuovi attributi, specificando alcuni dati fondamentali, tra cui:

- Un identificatore univoco dell'oggetto, nella forma di una sequenza numerica;
- Un nome, anch'esso univoco per l'oggetto;
- Un tipo di dato per l'oggetto;

- Una condizione di ricerca per l'oggetto.

L'eredità da attributi esistenti richiede soltanto di formalizzare nome e identificatore, per poi prelevare gli altri parametri da un tipo di attributo esistente. In tal caso, l'attributo ottenuto è definito SUP di quello esistente.

L'identificatore dell'oggetto deve rispettare una sintassi rigorosa basata sul concetto di OID (Object IDentfier). La sintassi prevede di utilizzare una stringa molto simile a quelle utilizzate nei sistemi di naming gerarchici in cui ogni elemento separato da punti esprime un grado si specializzazione. Il primo elemento, partendo da sinistra, è la radice o l'organizzazione di riferimento. Procedendo verso destra possono essere utilizzati annidamenti gerarchici di lunghezza arbitraria. La componente identificativa dell'organizzazione può essere scelta arbitrariamente soltanto se si limita l'utilizzo dello schema all'interno della propria infrastruttura. Se si desidera esportare lo schema per usi esterni la gerarchia deve essere assegnata presso la IANA[i], per garantire che l'organizzazione di riferimento sia univocamente identificata. In questo modo, se per esempio una data organizzazione è identificata dall'OID 1.1, le diramazioni interne possono essere:

Tabella 3.2: esempio di diramazioni di un DIT LDAP

OID	***Descrizione***
1.1	OID dell'Organizzazione
1.1.1	Elementi SNMP (rete)
1.1.2	Entry LDAP
1.1.2.1	Tipi di attributo LDAP
1.1.2.1.1	Attributo personalizzato
1.1.2.2	objectClass LDAP
1.1.2.2.1	objectClass Personalizzata

[i] La Internet Assigned Numbers Authority (IANA) è un organo preposto all'assegnazione di indirizzi ip e nomi riconosciuti universalmente. Le compagnie devono registrare indirizzi e nomi presso questo ente per una corretta integrazione con la rete Internet. Maggiori informazioni presso http://www.iana.org/

Questa struttura è utilizzata sia per l'identificazione di attributi e objectClass che per la caratterizzazione dei tipi di dato degli attributi, anch'essi rappresentati come entry nel sistema LDAP. L'esempio riportato fa uso della stessa nomenclatura anche per indicare elementi esterni a LDAP in senso stretto, come elementi per la gestione del traffico di rete. Questo è possibile grazie alla natura standardizzata del sistema di identificazione.

Il nome dell'attributo dovrebbe rispecchiare la funzione. Le raccomandazioni formulate dal team di OpenLDAP suggeriscono di determinare un prefisso significativo dell'organizzazione, tanto più preciso (quindi lungo) quanto più piccola è l'infrastruttura da modellare. Questo prefisso dovrebbe essere allegato a tutti i nomi degli attributi, il resto del nome dovrebbe essere specifico per l'attributo e fortemente identificativo della funzione. E' possibile anche specificare un commento da allegare a objectClass e attributi, tuttavia questo campo non è accessibile in tutte le implementazioni dei client LDAP, per le quali il nome parlato risulta più efficace. Questa combinazione tende a ridurre i potenziali conflitti sul nome in caso di schema distribuito.

Ogni tipo di dato ha un OID che segue le stesse regole di identificazione di attributi e objectClass. I tipi di dato più comuni sono già codificati da percorsi standard. Questi attributi sono sufficienti a formalizzare qualsiasi tipo di schema, tuttavia possono essere estesi e nuovi tipi possono essere aggiunti.

Tabella 3.3: tipi base degli attributi in OpenLDAP

Nome	*OID*	*Descrizione*
boolean	1.3.6.1.4.1.1466.115.121.1.7	Valore booleano.
directoryString	1.3.6.1.4.1.1466.115.121.1.15	Stringa in unicode (UTF-8)
distinguishedName	1.3.6.1.4.1.1466.115.121.1.12	DN LDAP
integer	1.3.6.1.4.1.1466.115.121.1.27	Intero
numericString	1.3.6.1.4.1.1466.115.121.1.36	Stringa numerica
OID	1.3.6.1.4.1.1466.115.121.1.38	OID
octetString	1.3.6.1.4.1.1466.115.121.1.40	Dato binario

Il tipo di dato “octetString” può essere utilizzato per memorizzare contenuti binari come immagini, è il caso dell’attributo standard jpegPhoto. Questo tipo di dato non è stato utilizzato nella progettazione del database.

Infine, le condizioni di ricerca sono utilizzate per definire il comportamento del sistema quando confronta un filtro con l’attributo su cui viene applicato. I tipi di ricerca possibili sono:

- Uguaglianza (equality): confronto dell’intero lemma con il valore dell’attributo;
- Inclusione (substring): per stringhe, si tratta di un’uguaglianza imperfetta;
- Ordinamento (ordering): per interi o sequenze di interi, consente di determinare la sequenza numerica tra due elementi, quindi di stabilire quale viene prima nell’ordine. Utile per query in la condizione del filtro verifica l’inclusione in un range di valori.

Le condizioni sono ulteriormente specificate dal comportamento rispetto alle maiuscole/minuscole (case sensitive o case insensitive) e rispetto alla presenza di spazi. Ciasuna combinazione condizione-comportamento è etichettata e può essere specificata per un singolo attributo.

Tabella 3.4: condizioni di match per attributi in OpenLDAP

Nome	***Tipo***	***Descrizione***
booleanMatch	equality	boolean
caseIgnoreMatch	equality	case insensitive, space insensitive
caseIgnoreOrderingMatch	ordering	case insensitive, space insensitive
caseIgnoreSubstringsMatch	substrings	case insensitive, space insensitive
caseExactMatch	equality	case sensitive, space insensitive
caseExactOrderingMatch	ordering	case sensitive, space insensitive
caseExactSubstringsMatch	substrings	case sensitive, space insensitive
distinguishedNameMatch	equality	distinguished name

Nome	*Tipo*	*Descrizione*
integerMatch	equality	integer
integerOrderingMatch	ordering	integer
numericStringMatch	equality	numerical
numericStringOrderingMatch	ordering	numerical
numericStringSubstringsMatch	substrings	numerical
octetStringMatch	equality	octet string
octetStringOrderingStringMatch	ordering	octet string
octetStringSubstringsStringMatch	ordering	octet string
objectIdentiferMatch	equality	object identifier

3.3. Il servizio di directory del progetto Bellerofonte

Le nozioni fornite fino a questo punto sono determinanti nella comprensione dei concetti fondamentali sulla realizzazione di nuovi schema. I rimanenti concetti rilevanti saranno trattati nelle seguenti sezioni e contestualizzati nelle specifiche del servizio di directory del progetto Bellerofonte.

3.3.1. Entità del database

La funzione primaria del database è memorizzare dati personali degli utenti. L'intera base dati può essere inoltre utilizzata a supporto delle procedure di assegnazione delle password e gestione delle tessere bibliotecarie. Le entità memorizzate sono pertanto:

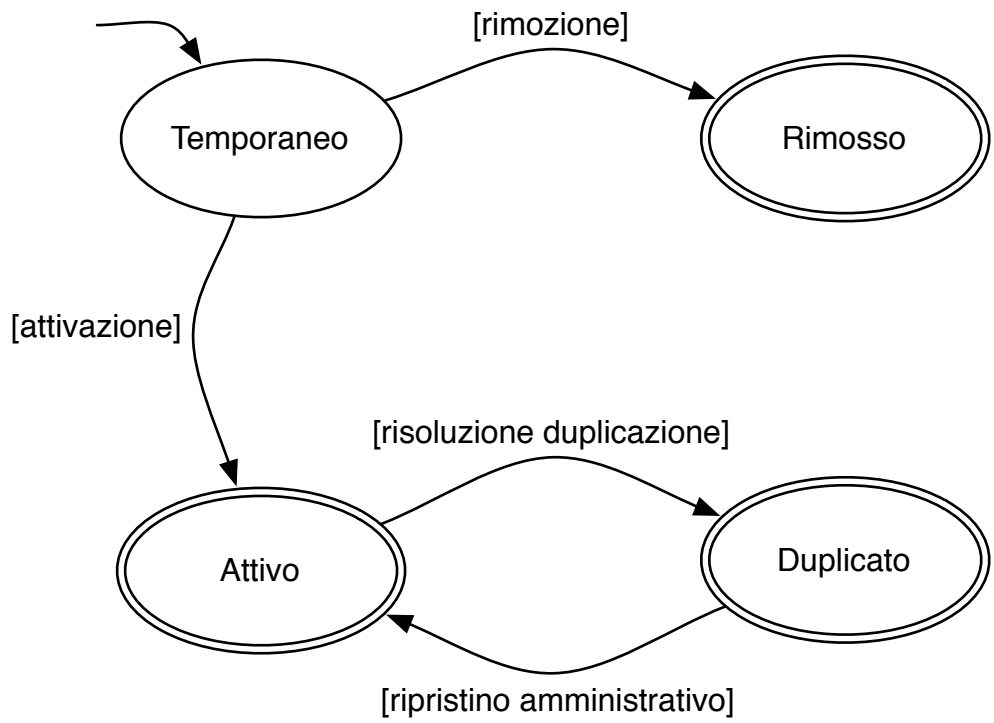
- Utenti. Queste entità modellano sia i dati personali degli utenti sia le informazioni necessarie alla gestione dei servizi. Gli utenti entrano nel sistema con un identificativo provvisorio e il loro profilo deve essere confermato perché possano operare correttamente con i servizi. Una volta confermati, gli utenti non possono più essere cancellati ma soltanto spostati. Tra gli eventi che causano lo spostamento delle entità utenti c'è la risoluzione di duplicati. Gli utenti non ancora confermati vengono cancellati in tal caso;

- Tessere bibliotecarie. Le tessere bibliotecarie sono create in base all'identificativo di iscrizione al prestito con cui l'utente viene importato nel sistema. La distinzione delle informazioni relative alle tessere ha lo scopo di identificare i casi di duplicazione in base al numero di tessera. Ogni tessera memorizza un identificativo per ogni proprietario noto e ogni utente memorizza l'identificativo dell'ultima tessera nota. In questo modo, dato un utente è possibile risalire immediatamente alla sua tessera e a tutti i proprietari oltre a lui presenti in associazione. La presenza di più di un proprietario costituisce un'anomalia definita duplicazione per numero tessera;
- Schede con password. Le schede sono la controparte informatica delle buste che vengono consegnate agli utenti. Quando gli operatori assegnano una busta il sistema risale all'entità che la descrive, dalla quale preleva la password e la assegna al profilo dell'utente. L'entità scheda deve essere invalidata dopo l'assegnazione. Il database è stato popolato con un cospicuo numero di schede per far fronte a una forte richiesta di abilitazioni;
- Biblioteche e account. Per ciascuna biblioteca vengono memorizzati alcuni dati di localizzazione fondamentali. Le entità biblioteche sono inoltre contenitori di altre entità: gli account per i servizi. Gli account per i servizi sono entità molto semplici utilizzate solamente per autenticare i servizi che utilizzano la base dati. Tali entità sono costituite da dati minimi richiesti dalla loro objectClass strutturale e da informazioni per l'autenticazione, quali uid e userPassword. Servizi come Proxy o altri che necessitano di avere accesso in lettura alla base dati hanno una controparte nella sezione servizi. Questi account sono suddivisi per biblioteca di provenienza.

Utenti

Si è scelto di non eliminare mai utenti che sono stati abilitati all'accesso ai servizi per mantenere sempre un riferimento ad azioni che possono essere state intraprese nel passato, memorizzate presso log o altro. Gli utenti sono memorizzati in due diversi rami, la risoluzione di un duplicato comporta lo spostamento di una entry dal ramo utilizzato per gli utenti attivi a quello utilizzato per gli utenti duplicati.

Figura 3.1: State Chart per l'entità utente



Gli utenti vengono inseriti nel sistema in stato Temporaneo. In questo stato non sono profili utilizzabili per l'accesso ai servizi. In caso di risoluzione di un caso di duplicazione a sfavore di un'entità temporanea, questa viene cancellata. Avvengono cancellazioni anche a seguito di procedure di manutenzione utilizzate per limitare il proliferare di entry non utilizzate.

Nel momento in cui un operatore verifica i dati di un utente e li conferma, l'entità passa in stato Attivo. Si suppone che le entità permangano in questo stato per la maggior parte della vita nel sistema. Dallo stato attivo è possibile spostarsi soltanto in stato Duplicato. La procedura di "ripristino amministrativo" consente di recuperare una entry spostata tra gli utenti duplicati. Questa procedura è definita amministrativa perché non consentita agli operatori del sistema.

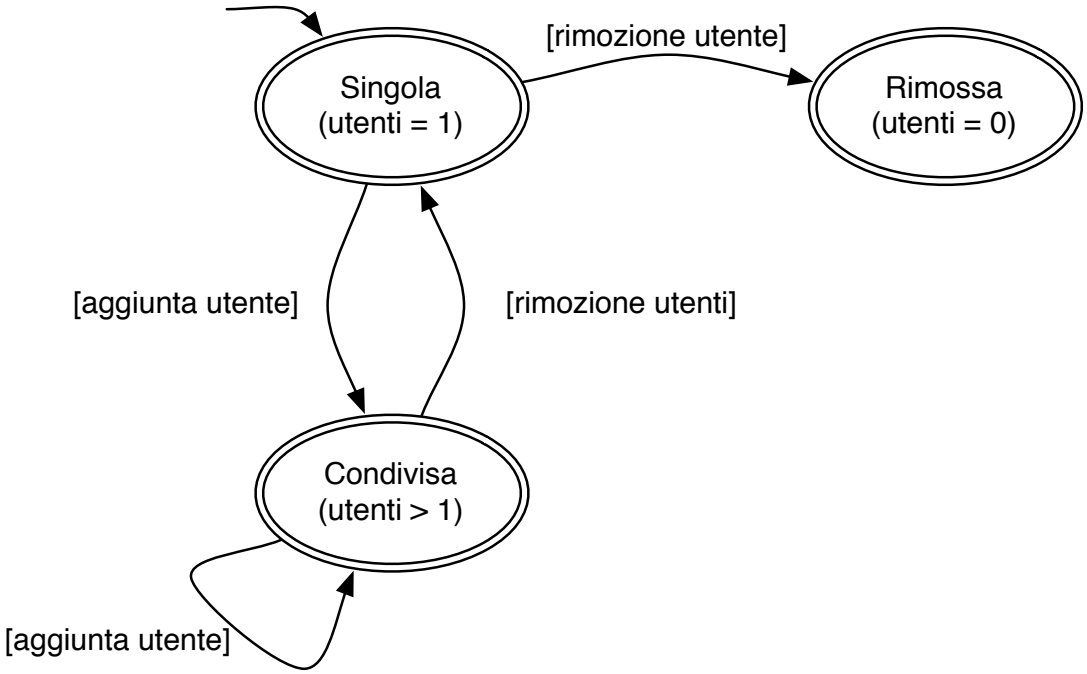
Le entità utenti possono permanere nello stato Attivo o Duplicato in modo stabile. Lo stato Temporaneo non può essere occupato in modo stabile: o una entry viene approvata op-

pure viene rimossa. Lo stato Rimosso è occupato da entry temporanee cancellate e costituisce l'unica uscita possibile dal sistema.

Tessere bibliotecarie

All'interno del sistema informativo preesistente non esiste la duplicazione di tessere, pertanto non è possibile emettere due tessere (intese come oggetti da consegnare agli utenti) con lo stesso numero. Anche l'associazione di un utente con una tessera è univoca, tuttavia è possibile modificare qualsiasi attributo di un profilo utente esistente. Ciò significa che è possibile "riassegnare" una tessera modificando tutti gli attributi dell'utente assegnatario ottenendo un profilo totalmente diverso eppure facente capo allo stesso identificativo di tessera. Questa procedura, sebbene sconsigliata, viene utilizzata in alcune circostanze dal personale della biblioteca. Per far fronte alle possibili problematiche di duplicazione che tale comportamento introduce, la tessera è stata distaccata dalle proprietà dell'utente in un'entità separata, in grado di memorizzare tutti gli utenti a cui risulta associata per quanto il sistema è in grado di percepire dalle entry. La tessera si considera "condivisa" quando è associata a più di un utente.

Figura 3.2: State Chart per l'entità tessera

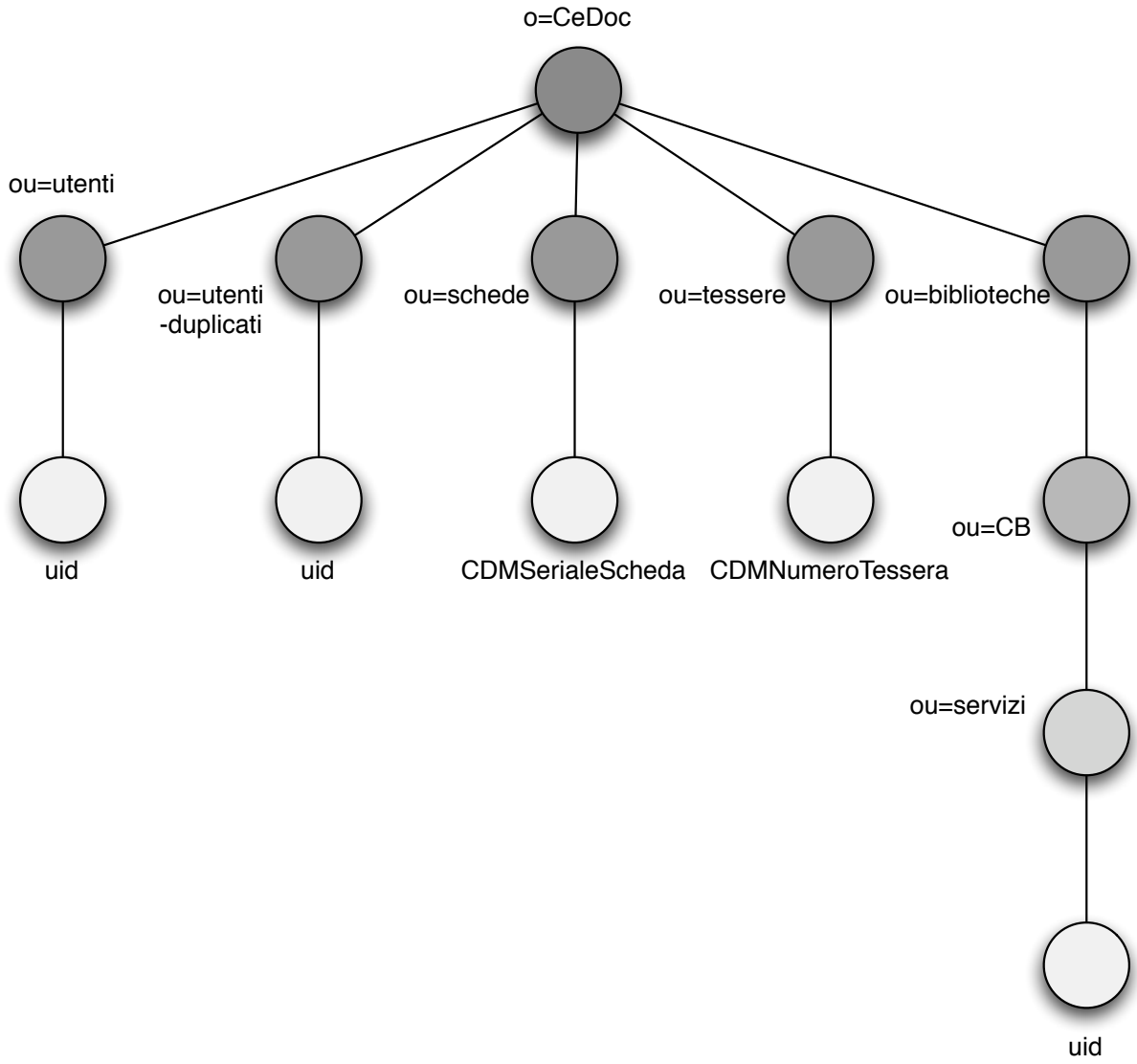


Le entità possono occupare tre stati: Singola, Condivisa e Rimossa. La rimozione equivale alla cancellazione dal sistema. La permanenza in stato Singola modella il caso più comune in cui a una tessera è associato un solo utente. In questo stato il sistema non incontra ambiguità e considera l'utente associato come istanza univoca. Se il numero di utenti associati aumenta il sistema incontra l'anomalia descritta in precedenza e tende a considerare gli utenti associati alla stessa tessera come dei duplicati. Questa anomalia deve essere risolta scatenando la notifica di un duplicato. La risoluzione del duplicato ha come effetto l'immediata rimozione dell'associazione con tutti gli utenti tranne quello favorito dalla risoluzione.

3.3.2. Struttura del DIT

Il Directory Information tree formulato per modellare il database fa capo a una singola radice, un'istanza di tipo organization denominata CeDoc. Tutti DN delle entry del database termineranno con la dicitura o=CeDoc. Adiecenti alla radice sono presenti le ramificazioni per modellare le categorie di elementi. Una ramificazione è stata proposta per ciascuna categoria.

Figura 3.3: Schema completo del DIT LDAP



Al disotto dell'elemento o=CeDoc sono presenti alcune organizational unit, componenti di primo livello dell'infrastruttura. Queste entry definiscono i rami utilizzati per utenti, utenti duplicati, schede, tessere e biblioteche. Rispettivamente, le entry di primo livello delle organizational unit (ou) saranno contraddistinte dai DN:

- ou=utenti,o=CeDoc;
- ou=utenti-duplicati,o=CeDoc;
- ou=schede,o=CeDoc;

- ou=tessere,o=CeDoc;
- ou=biblioteche,o=CeDoc.

Al successivo livello di diramazione, tutti i rami tranne biblioteche hanno direttamente entry foglia, ovvero entry che caratterizzano oggetti singoli non contenitori, il vero valore informativo del database. Il DN di entry per ciascun tipo sarà:

- uid=<id_utente>,ou=utenti,o=CeDoc per gli utenti attivi;
- uid=<id_utente>,ou=utenti-duplicati,o=CeDoc per gli utenti duplicati;
- CeDocMoSerialeScheda=<seriale>,ou=schede,o=CeDoc per le schede con password;
- CeDocMoNumeroTessera=<numero>,ou=tessere,o=CeDoc per le tessere bibliotecarie.

Gli attributi che cominciano con CeDocMo sono stati definiti per questo progetto e saranno descritti nella sezione successiva.

L'unica eccezione alle foglie alla seconda diramazione è il ramo biblioteche, per il quale sono definiti ulteriori livelli di diramazione, nella fattispecie:

- ou=<codice_biblioteca>,ou=biblioteche,o=CeDoc per ciascuna biblioteca di cui è noto il codice;
- ou=servizi,ou=<codice_biblioteca>,ou=biblioteche,o=CeDoc nel caso la biblioteca fornisca servizi che necessitano di autenticazione su LDAP;
- uid=<id_servizio>,ou=servizi,ou=<codice_biblioteca>,ou=biblioteche,o=CeDoc per ciascun servizio che richiede autenticazione nell'ambito della biblioteca.

In questo caso, sebbene ogni entry figlia di ou=biblioteche sia dichiarata come organizationalUnit, sono stati specificati alcuni attributi per definire le caratteristiche informative del "contenitore biblioteca".

3.3.3. Classi

Lo schema formulato modella attributi e classi che seguono le direttive sulla nomenclatura riportate in precedenza. Tutti gli elementi sono etichettati dal prefisso CeDocMo per essere riconosciuti immediatamente come attributi personalizzati. La codifica degli OID segue una sintassi valida, tuttavia gli OID riportati non sono coerenti con il sistema internazionale di assegnazione, per il quale non è ancora disponibile una codifica di base assegnata dalla IANA.

Lo schema definisce le seguenti objectClass:

- CeDocMoPersona, utilizzata per definire la struttura di memorizzazione dei dati personali. Ogni utente è caratterizzato da questa objectClass;
- CeDocMoAccount, utilizzata per completare i profili degli utenti con informazioni relative all'account (come password, abilitazioni ecc.). Ogni utente è definito anche da questa objectClass;
- CeDocMoScheda, utilizzata per definire la totalità degli attributi delle buste con password;
- CeDocMoTessera, utilizzata per definire gli attributi delle tessere bibliotecarie;
- CeDocMoBiblioteca, objectClass che definisce le caratteristiche informative degli elementi figli di ou=biblioteche;
- CeDocMoSrvAccount, account di servizio, utilizzato per definire le caratteristiche dei servizi associati a ogni biblioteca.

CeDocMoPersona

La definizione dei questa objectClass nella sintassi di schema OpenLDAP è la seguente:

```
objectclass ( 2.5.6.201 NAME 'CeDocMoPersona' SUP inetOrgPerson
      DESC 'Identificativo totale di una CeDocMoPersona'
      MUST ( CeDocMoDataNascita $ CeDocMoLuogoNascita $ CeDocMoSesso $ CeDocMoTipoDocumento $
CeDocMoNumeroDocumento $ CeDocMoIndirizzoResidenza $ CeDocMoCittaResidenza )
      MAY ( CeDocMoIndirizzoDomicilio $ CeDocMoCittaDomicilio ))
```

La classe è SUP di inetOrgPerson, una delle objectClass predefinite e maggiormente utilizzate. Questo significa che ogni entità dichiarata istanza della classe CeDocMoPersona può avere valori per tutti gli attributi definiti da inetOrgPerson. La scelta, come tutte le altre in questo senso, punta a favorire l'interazione con sistemi esistenti utilizzando per quanto possibile attributi con tipi e nomi noti. I restanti attributi sono stati formulati specificamente per cedoc.schema.

CeDocMoPersona è caratterizzata da un OID (2.5.6.201) che la identifica univocamente, da un nome (NAME) e da una descrizione (DESC). Inoltre, a parte l'eredità da inetOrgPerson (SUP), alcuni attributi sono definiti MUST e altri MAY. Gli attributi definiti MUST devono avere valore non nullo per tutte le istanze di CeDocMoPersona, gli attributi definiti MAY possono non avere valore.

La classe CeDocMoPersona si pone nella catena di ereditarietà che da inetOrgPerson porta alla classe strutturale person. La catena è CeDocMoPersona SUP inetOrgPerson SUP organizationalPerson SUP person. Queste eredità consentono di utilizzare i numerosi attributi definiti in tutte le classi della catena. In particolare, CeDocMoPersona utilizza gli attributi:

- uid: genericamente considerato identificatore univoco di un'entità. La maggior parte dei sistemi progettati per interagire con database LDAP fa uso di questo attributo. Attributo analogo è userid, non utilizzato in questo caso. La maggior parte dei DN per entità foglia in OpenLDAP ha come primo elemento il valore di uid;
- userPassword: attributo comune a molti schema LDAP. Al pari di uid viene utilizzato per procedure di autenticazione e autorizzazione;
- givenName: equivale al nome con cui un'entità è nota. Nel caso di entità che modellano persone, givenName è il nome proprio;
- cn: common name, cn è il nome completo di un'entità. Nel caso di entità che modellano persone, cn è il nome completo della persona: nome e cognome;
- sn: surname. Si tratta del cognome per entità che modellano persone;
- mail: indirizzo e-mail primario della persona. Tutti gli schema che offrono credenziali per la gestione di servizi di posta elettronica utilizzano mail come punto di partenza. Di fatto si tratta di una stringa e può essere utilizzato a semplice valore informativo. La corretta

compilazione del campo mail offre opportunità di estendere le funzionalità della base dati in caso di aggiunta di servizi relativi alla posta elettronica.

Nel corso della progettazione della classe si è scelto di modellare gli obblighi legali di memorizzazione dei dati, rendendo MUST tutti gli attributi che devono aver valore per considerare un'anagrafica completa. Sono dichiarati MAY attributi suppletivi alla caratterizzazione dell'anagrafica. Questa scelta costringe a gestire l'obbligo di avere valori non nulli per alcuni attributi anche quando non esiste un valore vero e proprio. La logica necessaria per la valutazione di ogni valore inserito dagli utenti è delegata all'Interfaccia Web, la quale non memorizza profili ritenuti validi se non completi di valore per tutti gli attributi MUST.

Tabella 3.5: CeDocMoPersona

Nome	***Tipo***	***Descrizione***
CeDocMoDataNascita	Stringa Numerica (8L)	Data di nascita (YYYYMMDD)
CeDocMoLuogoNascita	Stringa	Luogo di nascita (Città)
CeDocMoSesso	Stringa (1L)	Sesso (una lettera M o F)
CeDocMoTipoDocumento	Stringa	Tipo di documento (Carta di identità, passaporto o patente)
CeDocMoNumeroDocumento	Stringa (20L)	Numero del documento (massimo 20 lettere)
CeDocMoIndirizzoresidenza	Stringa	Indirizzo di residenza
CeDocMoCittaResidenza	Stringa	Città di residenza
CeDocMoIndirizzoDomicilio	Stringa	Indirizzo di domicilio
CeDocMoCittaDomicilio	Stringa	Città di domicilio

Le celle con sfondo colorato riportano gli attributi MAY.

In questo caso, come in tutte le occorrenze di attributi che memorizzano date, si è scelto di utilizzare una notazione seriale con le quattro cifre dell'anno, le due cifre del mese e le due cifre del giorno, con cifre 0 (zero) a completamento degli spazi vuoti. Questo consente di dichiarare la data come una stringa di interi e applicare facilmente condizioni di ordinamento, rendendo possibile l'esclusione o inclusione di risultati direttamente tramite i filtri della query LDAP.

Il sesso è memorizzato attraverso una stringa di una lettera, che può ospitare i valori M e F, semplici da gestire come un valore booleano.

I restanti attributi sono prevalentemente stringhe di lunghezza predefinita, ad eccezione del numero di documento per il quale è espresso un limite in lunghezza in 20 caratteri.

CeDocMoAccount

La classe CeDocMoAccount definisce le caratteristiche di ciascun utente per quello che riguarda l'account, ovvero la componente del profilo che ha a che fare con i servizi. La separazione degli attributi nelle due classi classi (persona e account) favorisce la chiarezza. Nel momento in cui è necessario applicare modifiche è sufficiente determinare a che tipo di sfera semantica appartengono (se all'account o ai dati personali) per facilitare la localizzazione della parte da modificare. Questa scelta costringe a dichiarare ogni entità utente come istanza sia di CeDocMoPersona sia di CeDocMoAccount, operazione comune nel sistema OpenLDAP.

La classe che modella l'account è dichiarata come segue:

```
objectclass ( 2.5.6.202 NAME 'CeDocMoAccount' SUP posixAccount AUXILIARY
        DESC 'Identificativo CeDocMoAccount LDAP'
        MUST ( CeDocMoUltimaTessera $ CeDocMoUltimoServizio $ CeDocMoDataUltimoCambio $ CeDoc-
MoDataUltimaAttivazione $ C
eDocMoResponsabile )
        MAY ( CeDocMoServizi $ CeDocMoGarante $ CeDocMoListaMinori $ mail $ mailAlternateAd-
dress $ mailMessageStore ))
```

La classe eredita gli attributi da un'altra classe molto utilizzata nella definizione di basi dati LDAP: posixAccount. posixAccount è utilizzata frequentemente quando è necessario

memorizzare informazioni per autorizzazione e autenticazione a sistemi Unix. Nella fattispecie gli attributi ereditati da gestire in ogni profilo sono:

- uidNumber: un intero, generalmente utilizzato per identificare un utente in un dominio amministrativo;
- gidNumber: identifica il gruppo di appartenenza nella logica a gruppi dei sistemi Unix;
- homeDirectory: home directory degli utenti nei sistemi Unix.

Si noti che i tre attributi sono dichiarati MUST per posixAccount ma non sono necessari al funzionamento del sistema. Casi di questo genere possono essere gestiti proponendo ugualmente un valore per gli attributi anche se privo di significato. I valori possono essere semplicemente modificati nel caso diventino utili in futuro. La scelta di ereditare da posixAccount la struttura della objectClass si giustifica nella possibilità di estendere i profili utente per l'utilizzo combinato con servizi gestiti da sistemi Unix.

CeDocMoAccount segue la stessa logica di CeDocMoPersona, imponendo il completamento di tutti i parametri necessari alla definizione minima di un account. I rimanenti attributi, specificati in MAY, definiscono parametri aggiuntivi che un account può non avere. L'attributo CeDocMoServizi mantiene traccia di tutti i servizi a cui un utente è autorizzato ad accedere. Un valore nullo per questo attributo appare insensato, poiché la funzione dell'account LDAP è quella di accedere ad almeno un servizio, tuttavia la formulazione è corretta perché anche la prima attivazione è soggetta a verifica. Le nuove entità entrano nel database senza poter essere utilizzate per l'accesso ad alcun servizio.

La classe CeDocMoAccount è dichiarata AUXILIARY, ovvero è amministrativamente considerata una classe non strutturale. Un'entità non può essere dichiarata istanza della sola classe CeDocMoAccount poiché necessita degli attributi di una classe strutturale. La struttura è in questo caso offerta dalla classe person, da cui eredita CeDocMoPersona.

Tabella 3.6: CeDocMoAccount

Nome	*Tipo*	*Descrizione*
CeDocMoUltimaTessera	Stringa	Ultima tessera bibliotecaria nota
CeDocMoUltimoServizio	Stringa (5L)	Servizio da cui proviene l'ultima tessera nota (AURIGA o SEBINA)
CeDocMoDataUltimoCambio	Stringa Numerica (8L)	Data di ultimo cambio password
CeDocMoDataUltimaAttivazione	Stringa Numerica (8L)	Data di ultima verifica dei dati
CeDocMoResponsabile	Stringa	uid dell'operatore che ha eseguito l'ultima verifica dei dati nota.
CeDocMoServizi	Stringa (multipla)	Servizi abilitati per l'utente
CeDocMoGarante	Stringa	Se l'utente è un minore, uid del garante (genitore)
CeDocMoListaMinori	Stringa (multipla)	Se l'utente è garante di minori, lista degli uid di questi

Gli attributi che si riferiscono all'ultima tessera e all'ultimo servizio riguardano la natura delle informazioni migrate dalle anagrafiche precedenti. L'ultima tessera è il numero di tessera (completo dei prefissi di zona) associata al profilo nell'anagrafica dalla quale è stata derivata l'ultima modifica all'utente. Le entità del ramo tessere conservano associazione agli utenti loro proprietari, tuttavia nell'entità utente è conservata nozione solo dell'ultima tessera di cui l'utente si è servito. La nozione sull'ultimo servizio consente di evidenziare da quale delle due anagrafiche è stato importato o modificato il profilo. Questa informazione ha valore in sede di manutenzione.

Le due date, ultimo cambio e ultima attivazione, memorizzano il momento in cui l'utente rispettivamente cambia la propria password e si presenta per la validazione dei dati personali.

Il sistema mantiene sempre traccia dell'operatore che ha eseguito il salvataggio dei dati dell'utente per l'ultima volta. Memorizzare questa informazione ha due valenze: consente agli operatori di rintracciare il collega che ha effettuato le modifiche in caso di dubbi e consente agli amministratori di tracciare il responsabile di una validazione non corretta.

Garante e ListaMinori intervengono nella gestione degli utenti minorenni. Per associare al genitore la responsabilità delle azioni del minore è necessario memorizzare un identificativo del genitore presso il profilo del figlio. Inoltre, se un adulto è associato nel profilo di un minore, l'identificativo del figlio viene memorizzato nella lista dei minori associati. Per questo motivo CeDocMoListaMinori è un attributo multiplo. Questo dato ha valore prevalentemente informativo.

CeDocMoScheda

La classe CeDocMoScheda è strutturale ed elenca attributi necessari alla gestione delle buste con password. La classe è formulata nel modo seguente:

```
objectclass ( 2.5.6.203 NAME 'CeDocMoScheda'
      DESC 'Collezione di attributi per le schede password'
      MUST ( CeDocMoSerialeScheda $ CeDocMoPasswordScheda )
      MAY ( CeDocMoDataConsegnaUtente $ CeDocMoDataConsegnaBiblioteca $ CeDoMoIdDestinatario $
CeDocMoIdOperatoreScheda $ CeDocMoBibliotecaDestinataria )
```

La logica di formulazione di questa classe prevede in MUST gli attributi caratteristici della scheda, ovvero il seriale univoco e la password associata; senza questi attributi un'entità scheda non ha utilità. Gli attributi in MAY competono alla gestione del percorso della scheda nel sistema, attraverso la procedura di assegnazione a un utente.

Tabella 3.7: CeDocMoScheda

Nome	*Tipo*	*Descrizione*
CDMSerialeScheda	Stringa Numerica (8L)	Seriale stampato sulla scheda, 8 lettere
CDMPasswordScheda	Stringa (8L)	Password stampata dentro la scheda, 8 lettere
CDMDataConsegnaUtente	Stringa Numerica (8L)	Data di consegna all'utente (se è NULL la scheda non è stata ancora consegnata)
CDMDataConsegnaBiblioteca	Stringa Numerica (8L)	Data di consegna alla biblioteca del plico contenente la scheda
CDMIdDestinatario	Stringa	uid dell'utente destinatario della scheda
CDMIdOperatoreScheda	Stringa	Operatore che assegna la scheda all'utente, uid
CDMBibliotecaDestinataria	Stringa	Biblioteca destinataria del plico contenente la scheda

CeDocMo è stato abbreviato in CDM.

Il sistema utilizza gli attributi di questa classe per gestire le procedure di consegna delle schede a una biblioteca e di consegna di una singola scheda a un utente. Per consegna alla biblioteca si intende la registrazione di un plico di schede ordinate di cui è noto il primo e l'ultimo seriale. A scopo informativo tutti i seriali compresi vengono registrati come consegnati alla biblioteca. La consegna all'utente conivolge una scheda alla volta e registra il destinatario della scheda (IdDestinatario) e colui che la assegna (IdOperatoreScheda). Queste informazioni sono di particolare valore per determinare se una scheda è già stata assegnata e a chi, in caso di segnalazioni di problemi tecnici.

CeDocMoTessera

La classe CeDocMotessera è utilizzata per modellare entità per tracciare il percorso delle tessere bibliotecarie all'interno del sistema. Le istanze di questa classe sono le uniche a poter essere cancellate completamente nel corso delle normali procedure automatizzate del sistema. La classe è strutturale ed è dichiarata come segue:

```
objectclass (2.5.6.204 NAME 'CeDocMoTessera'
      DESC 'Classe per modellare le tessere importate da Auriga e Sebina'
      MUST ( CeDocMoNumeroTessera $ CeDocMoNomeServizioTessera $ CeDocMoProprietarioTessera ))
```

Gli attributi presenti sono dichiarati tutti MUST. Essi compongono ogni entità possibile modellata da questa classe poiché costituiscono informazioni obbligatorie per la distinzione di un profilo di tessera bibliotecaria.

L'attributo CeDocMoNumeroTessera è un identificatore univoco nel sistema. Le informazioni relative ai proprietari multipli sono memorizzate nell'attributo multiplo CeDocMoProprietarioTessera, un attributo multiplo designato a memorizzare un set di uid.

Il nome del servizio da cui proviene la tessera riporta un identificativo di quale delle due anagrafiche precedenti ha fornito la tessera.

CeDocMoBiblioteca

La classe modella le caratteristiche degli elementi biblioteca. Come accennato, questi elementi sono sia contenitori di altre entità sia entità con valore informativo. Gli attributi elencati dalla classe costituiscono il valore informativo e sono stati derivati dalle informazioni liberamente disponibili sulle biblioteche del polo provinciale. La classe è definita come segue:

```
objectclass ( 2.5.6.205 NAME 'CeDocMoBiblioteca'
        DESC 'Definizione della entry per il ramo biblioteche'
        SUP organizationalUnit
        MAY ( CeDocMoIndirizzoBiblioteca $ CeDocMoCittaBiblioteca $ CeDocMoNomeBiblioteca $
CeDocMoTelefonoBiblioteca $ CeDocMoResponsabileBiblioteca $ CeDocMoEmailBiblioteca ))
```

La natura delle entità istanze di questa classe è chiarificata dalla clausola di ereditarietà. Essendo SUP di organizationalUnit (ou), CeDocMoBiblioteca è definita da un diverso genere

di elementi strutturali, i quali la rendono adatta a fare da contenitore ad altre ou o a nodi foglia. I restanti attributi definiscono dettagli sulla biblioteca e sono stati completati con le informazioni disponibili.

Tabella 3.8: CeDocMoBiblioteca

Nome	*Tipo*	*Descrizione*
CDMIndirizzoBiblioteca	Stringa	Indirizzo della biblioteca
CDMCittaBiblioteca	Stringa	Città di collocazione della biblioteca
CDMNomeBiblioteca	Stringa	Nome esteso della biblioteca
CDMTelefonoBibilioteca	Stringa Numerica	Numero di telefono della biblioteca
CDMresponsabileBiblioteca	Stringa	Nome esteso del responsabile della biblioteca (non uid)
CDMEmailBiblioteca	Stringa	Indirizzo e-mail della biblioteca

Le informazioni rispecchiano quelle disponibili per gli utenti. Nel caso della classe biblioteca si è scelto di non utilizzare attributi predefiniti per mantenere una maggiore chiarezza espressiva. Il significato di ogni attributi è chiaro, tuttavia è opportuno notare che il nome del responsabile è un nome proprio e non è collegato a nessun utente specifico del sistema. Inoltre, l'indirizzo e-mail della biblioteca non è soggetto alle considerazioni fatte sull'estensione del progetto per la gestione della posta elettronica.

Il DN di ciascuna biblioteca è costituito dalla sigla univoca affidata dal Sistema Bibliotecario Nazionale e dal resto del precorso: ad esempio

```
ou=CD,ou=biblioteche,o=CeDoc
```

per la biblioteca interna del CeDoc.

CeDocMoSrvAccount

CeDocMoSrvAccount modella il generico account utilizzato dai servizi per accedere al database LDAP. In favore di sicurezza è opportuno disporre di gradi di accesso con possibilità

di interazione ridotte rispetto all'accesso amministrativo. Per evitare mescolanze con i veri e propri utenti del sistema si è scelto di realizzare account semplici direttamente inseriti nelle biblioteche. Sul ramo biblioteche è consentita l'autenticazione e i servizi fanno uso di credenziali di accesso definite per interrogare altri rami del DIT, ciascuno con la possibilità di restrizione (si veda la sezione relativa al controllo di accesso). L'account generico è così formulato:

```
objectclass ( 2.5.6.206 NAME 'CeDocMoSrvAccount'
        DESC 'Account per i server che accedono al db'
        SUP account
        MAY (userPassword))
```

la struttura è estremamente semplice e prevede un'eredità dalla classe account alla quel è aggiunto soltanto l'attributo userPassword. Questa struttura è la minima possibile per ottenere un account basato su una classe strutturale e dotato di password.

3.3.4. Attributi

La maggior parte degli attributi riportati è ha valore informativo o di catalogo, sia per dati da memorizzare obbligatoriamente sia per dettagli aggiuntivi. Questo tipo di attributi non definisce caratteristiche particolari per il tipo di dato: per lo più si tratta di eredità dall'attributo predefinito name. Gli attributi con valenza tecnica, ovvero utilizzati per la gestione del sistema, sono soggetti a condizioni sul tipo di dato e vincoli sulle modalità di ricerca. In questa sezione sono riportati alcuni attributi particolarmente significativi per le scelte compiute nella loro definizione.

Date

Un esempio di data, la data di nascita, è riportato di seguito:

```
attributetype ( 2.5.4.203 NAME 'CeDocMoDataNascita'
      DESC 'Data di nascita'
      EQUALITY numericStringMatch
      SUBSTR caseIgnoreSubstringsMatch
      ORDERING numericStringOrderingMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.36{8} )
```

Le date sono definite come stringhe di interi (da 0 a 9) con un numero di caratteri limitato a 8. La caratterizzazione delle date in forma serale (AAAAMMGG) consente di ordinare i valori di data e di determinare quale viene prima nell'ordinamento numerico senza ricorrere a una logica esterna per l'ordinamento alfabetico. In tutti i casi in cui le date vengono proposte agli utenti sono state implementate funzioni di formattazione per renderle leggibili in una notazione del tipo GG/MM/AAAA. Analogamente, il sistema è in grado di assorbire date formulate in più notazioni e di riformattarle nella notazione seriale.

In termini di caratteristiche dell'attributo, la scelta della serializzazione si traduce in una stringa numerica sulla quale non vengono formulati vincoli di particolare ristrettezza per quanto riguarda la ricerca con sottostringhe. La clausola EQUALITY impone un match completo quando si tratta di esprimere l'uguaglianza esatta con un lemma di ricerca. Infine, la clausola ORDERING consente l'ordinamento tra stringhe numeriche e favorisce query con filtri del tipo

```
data > CeDocMoDataNascita
```

Si fa particolare utilizzo di questa funzione per la gestione della disabilitazione degli utenti quando relativa al mancato rispetto delle scadenze. Memorizzare tali informazioni nel formato di data consente di formulare query che riportino soltanto gli utenti che hanno cambiato la password entro gli ultimi sei mesi e abbiano dati validi da meno di due anni. La query esclude automaticamente gli utenti che non soddisfano i requisiti legali, senza la necessità di procedere alla disattivazione degli account specificata attraverso un attributo. Nel momento in cui l'utente soddisfa la procedura richiesta, la data cambia e il profilo viene incluso nei risultati della query di autorizzazione a ogni richiesta.

Attributi multipli

Se non diversamente specificato, tutti gli attributi utilizzati in OpenLDAP sono multipli. Gli attributi multipli consentono di affidare molteplici valori ai tipi di attributo mantenendo un separazione concettuale precisa tra ciascun valore. Un esempio di forte impiego degli attributi multipli è CeDocMoServizi, definito come segue:

```
attributetype ( 2.5.4.222 NAME 'CeDocMoServizi'
      DESC 'Elenco degli uid-composti dei servizi abilitati'
      EQUALITY caseIgnoreMatch
      SUBSTR caseIngnoreSubstringMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} ))
```

La definizione dell'attributo è stata riportata in forma completa. In realtà CeDocMoServizi è SUP di uid e ne eredita tutte le caratteristiche. Per maggiore chiarezza l'attributo è stato definito con tutte le caratteristiche di uid.

L'attributo multiplo può assumere valori in una quantità arbitraria di stringhe da 256 caratteri ciascuna. Per definire l'attributo come valore singolo la sintassi corretta sarebbe stata:

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} SINGLE-VALUE
```

Nella logica del sistema, CeDocMoServizi ha valori tra i seguenti:

- Internet, per intendere l'abilitazione a internet;
- InternetAAAAMMGG, per intendere l'abilitazione a internet ma con navigazione interdetta fino alla data AAAAMMGG causa sospensione per violazione delle norme;
- OperatoreLdap, per intendere l'accesso come operatore al sistema.

L'attributo è facilmente estensibile a un dominio arbitrario di valori con significato simbolico per intendere l'abilitazione a servizi futuri. Al momento di attivazione del sistema di gestione posta elettronica basato su LDAP, al dominio di validità sarà aggiunto l'elemento Mail per intendere l'abilitazione a ricevere e inviare posta elettronica.

Questo approccio alla gestione dei servizi, benché facilmente estendibile, non è standard. Gestire l'interrogazione del database attraverso una query che verifica il valore di attributi aggiunti agli schema predefiniti può causare problematiche di interazione con servizi la cui componente di dialogo è legata a query fissate (ad esempio essendo in grado di interrogare solo uid e userPassword).

Esempi analoghi di utilizzo di attributi multipli sono CeDocMoListaMinori (per gli account) e CeDocMoIdProprietario (per le tessere bibliotecarie).

Nel primo caso l'attributo multiplo modella la possibilità di avere più utenti minorenni associati alla responsabilità di un singolo adulto. Nel secondo si vuole esprimere la possibilità di una tessera di appartenere a più di un utente, consentendo di identificare le anomalie di duplicazione.

3.4. Configurazione del servizio

3.4.1. Indici

In OpenLDAP gli indici hanno la stessa funzione che in tutte le strutture database: consentono di migliorare le prestazioni di ricerca. Realizzare indici è fortemente consigliato per rendere più veloci le interrogazioni su attributi molto significativi. La determinazione degli attributi significativi costituisce una considerazione progettuale, la scelta del tipo di indice è di carattere implementativo.

OpenLDAP mette a disposizione alcuni tipi di indici. Tra di essi, quelli utilizzati nel sistema sono:

- eq (uguaglianza), utilizzato per indicizzare attributi sui quali si richiede matching esatto per le query;
- pres, estende eq consentendo il matching con wildcard;
- sub (sottostringa), utilizzato per indicizzare attributi su cui viene fatto un matching parziale.

La sintassi di definizione degli indici prevede di specificare una lista di attributi ai quali applicare gli indici e una lista di tipi di indice su cui applicarli. E' possibile definire più di un tipo di indice per ciascun tipo di attributo, tuttavia è opportuno evitare gli indici inutili poiché la loro manutenzione impatta sulle prestazioni in inserimento e modifica.

Gli indici utilizzati nel sistema sono:

```
(1)   index objectclass,entryCSN,entryUUID                                  eq
(2)   index sn,givenName,cn                                                 eq,pres,sub
(3)   index uid,userPassword                                                eq,pres
(4)   index CeDocMoDataUltimoCambio,CeDocMoDataUltimaAttivazione            eq,pres,sub
(5)   index CeDocMoServizi                                                  eq,pres
(6)   index CeDocMoSerialeScheda                                            eq,pres,sub
(7)   index CeDocMoIdDestinatario,CeDocMoIdOperatoreScheda                  eq,pres
(8)   index CeDocMoNumeroTessera,CeDocMoProprietarioTessera                 eq,pres
```

le righe sono numerate per riferimento nella trattazione seguente.

Gli attributi della riga 1 sono indicizzati per favorire le prestazioni nelle procedure di sincronizzazione, trattate in seguito. La riga 2 definisce un'indicizzazione molto completa per gli attributi sui quali insistono maggiormente le ricerche eseguite dagli operatori nel corso delle procedure di gestione. La riga 3 offre un'indicizzazione generica per gli attributi usati in ogni richiesta di credenziali: è molto frequente cercare per uid. La riga 4 ospita l'indicizzazione delle date, utilizzate frequentemente per l''interrogazione da parte dei servizi. Sempre per quanto riguarda gli account, la riga 5 indicizza il parametro relativo ai servizi, di frequente interrogazione.

Per favorire la ricerca delle schede, eseguita tutte le volte che ne viene assegnata una, si è realizzato un indice sul seriale nella riga 6. La riga 7 non è attualmente utilizzata, tuttavia i dati sono indicizzati per favorire le prestazioni di un'interfaccia amministrativa di prossima realizzazione. Infine, la riga 8 offre gli indici utilizzati per la ricerca rapida delle tessere bibliotecarie, frequentemente associata alle ricerche degli utenti.

E' possibile specificare gli attributi da indicizzare con una sintassi più ristretta o una notazione progressiva, tuttavia in favore di chiarezza è stata scelta questa notazione.

3.4.2. Controllo di accesso

Il controllo di accesso in OpenLDAP utilizza un insieme di regole lette in sequenza, le quali specificano:

1. Su quale parte del DIT insiste la regola;
2. Opzionalmente, su quali attributi insiste la regola;

3. A chi si applica la regola, ovvero a quali credenziali di accesso si riferisce;
4. Che tipo di accesso viene garantito.

Le regole possono contenere altri dettagli.

Per ciascuna regola viene eseguito un confronto sulle condizioni in sequenza. Se il confronto ha esito positivo per tutte le direttive fino alla (3), viene garantito il diritto specificato nella direttiva (4). Se uno qualunque dei confronti fallisce, su passa alla regola successiva. Si suppone che l'elenco di regole termini implicitamente con la regola:

```
access to *
by * none
```

Ovvero, negare l'accesso su tutto il DIT a chiunque. Se viene raggiunta l'ultima regola significa che non è stata formulata alcuna direttiva che consenta l'accesso con le credenziali con cui ci si presenta al server. Se viene raggiunta una regola le credenziali sono considerate valide per il grado di accesso della sua condizione (4).

La sintassi del parametro (1) parte della specifica di un DN come base per localizzare un punto all'interno del DIT. In base al DN è possibile specificare clausole di estensione degli oggetti accessibili.

Tabella 3.9: clausole di espansione di un DN per la ricerca

Clausola	*Estensione*
dn.base	Nessuna estensione, solo il DN specificato.
dn.one	Solo i figli diretti del DN.
dn.children	Tutti i figli a qualsiasi livello tratte il DN stesso.
dn.subtree	Tutti i figli del DN e il Dn stesso.

Oltre alle clausole di estensione, la condizione (1) può essere arricchita da filtri sulle objectClass di cui si richiedono istanze, questi filtri non sono stati utilizzati all'interno del progetto.

Il parametro (2) è una semplice lista di attributi, intestata dalla parola chiave "attrs". In questa condizione è possibile specificare una restrizione dell'accesso a una cerchia di attributi definita. In questo senso è necessario fare attenzione all'ereditarietà tra attributi: se esiste una lista di attributi che rende uid accessibile, tutti gli attributi SUP di uid sono accessibili a loro volta.

La condizione (3) specifica a quali entità è applicata la regola e fa uso di simboli predefiniti o di una sintassi del tutto simile a quella della condizione (1).

Tabella 3.10: Regole di accesso

Parola chiave o simbolo	***Entità***
*	Tutte le entità, inclusi utenti anonimi e non autenticati.
anonymous	Utenti non autenticati (clausola usata per definire regole di autenticazione).
users	Tutti gli utenti che hanno superato il processo di autenticazione.
self	Solo l'utente identificato dalla condizione (1), il proprietario.
dn.<clausola estensione>	Sintassi come al punto (1).

Infine, la condizione (4) specifica che tipo di accesso garantire all'entità o alle entità identificate dalle condizioni (1) e (2).

Tabella 3.11: Livelli di accesso

Livello	***Descrizione***
none	Nessun accesso.

Livello	*Descrizione*
auth	Accesso minimo per autenticarsi (di solito utilizzato per uid e userPassword).
compare	Accesso per confronto.
search	Accesso per L'applicazione di filtri di ricerca.
read	Accesso alla lettura dei risultati della ricerca.
write	Accesso completo.

Ogni livello implica tutti i precedenti nell'ordine.

In base alle nozioni riportate in merito alla composizione delle regole di accesso (acl), si riportano e commentano alcune regole utilizzate nel sistema.

```
access to dn.children="ou=biblioteche,o=CeDoc" attrs=userPassword
 by anonymous auth
```

La regola consente a tutte le entità figlie a qualsiasi livello del DN ou=biblioteche,o=CeDoc di accedere al proprio attributo userPassword scopo di autenticazione. Sotto a questo DN sono presenti entry utilizzate per autenticare i servizi che fanno uso della directory e la regola consente loro l'accesso al sistema. Questa regola non sancisce alcun livello di accesso alle entry di cui hanno bisogno, funzione svolta da regole come la seguente:

```
access to dn.children="ou=utenti,o=CeDoc"
 attrs=uid, userPassword, CeDocMoDataUltimoCambio, CeDocMoDataUltimaAttivazione, CeDocMoServi-
zi, entry
 by dn.base="userid=proxy,ou=servizi,ou=CD,ou=biblioteche,o=CeDoc" read
```

La regola sancisce il diritto di lettura degli attributi considerati identificativi di un account CeDoc per tutte le entry del sottoramo ou=utenti,o=CeDoc. Questa regola garantisce il particolare i diritti all'utente userid=proxy,ou=servizi,ou=CD,ou=biblioteche,o=CeDoc, ovvero l'utente impiegato dal servizio Proxy per interrogare la base dati e ottenere dati sugli utenti.

Ulteriori servizi ai quali garantire lo stesso livello di accesso possono essere specificati come righe aggiuntive della stessa regola, identificate dalla parola chiave “by”.

Se due servizi hanno accesso allo stesso sottoramo ma uno deve spaziare su un maggior numero di attributi, devono essere specificate due regole: una dà accesso al sottoramo limitatamente agli attributi comuni a entrambi i servizi, l’altra dà accesso allo stesso sottoramo per gli attributi aggiuntivi solo al servizio che ha diritto a utilizzarli.

3.4.3. Replicazione

Per replicazione si intende la realizzazione di copie esatte della directory su più di un sistema. Una replica può essere utilizzata come copia attiva per il bilanciamento del carico, oppure può funzionare da backup da attivare in caso di emergenza. OpenLDAP mette a disposizione due diversi metoddi di replicazione:

- slurpd: si tratta di un sistema push. Un demone aggiuntivo ascolta le query di scrittura eseguite sulla directory principale (Master) e replica ciascuna query su tutte le basi alternative (slave). Il metodo è disponibile da diverso tempo in OpenLDAP ma risulta poco efficiente per due motivi: non utilizza le elevate prestazioni in lettura del sistema per rendere più veloce la replica e si limita a replicare query, quindi replica anche modifiche consecutive a una stessa entry svoltesi in intervalli in cui la entry non sarebbe stata altrimenti utilizzata, rendendo la prima replica inutile.
- syncrepl. Il nuovo sistema di replicazione di OpenLDAP opera con un sistema pull: non richiede altri demoni attivi, si tratta semplicemente di una query persistente che ciascuno slave (detto consumer) mantiene sul master (detto provider), rinfrescata a intervalli definiti. Ad ogni intervallo ciò che è cambiato viene replicato sul consumer. Il sistema è efficiente perché si basa sulla ricerca che in LDAP è molto rapida ed è inerentemente fault tolerant perché non richiede che le modifiche siano recepite in ordine e continuamente.

Nel corso del progetto la base dati è stata configurata per utilizzare syncrepl. La directory principale, bersaglio di tutte le attività di scrittura, richiede una semplice configurazione:

```
overlay syncprov
```

```
syncprov-checkpoint 10 2
syncprov-sessionlog 10
```

La prima riga dichiara il sistema come syncrepl provider, la secondo determina il limite minimo per considerare la base dati modificata (numero di query, 10 o secondi, 2). La terza riga specifica la dimensione massima del file di log per ogni sessione di sincronizzazione (in KB).

Ciascun consumer deve essere configurato con una procedura di sincronizzazione per ogni provider a cui si collega. E' possibile specificare un numero arbitrario di procedura di sincronizzazione, tuttavia più procedure per lo stesso provider non sono mai utilizzate. Molteplici procedure per più provider diversi sono accettabili solo se vertono su parti del DIT non sovrapposte. Una generica configurazione di consumer è:

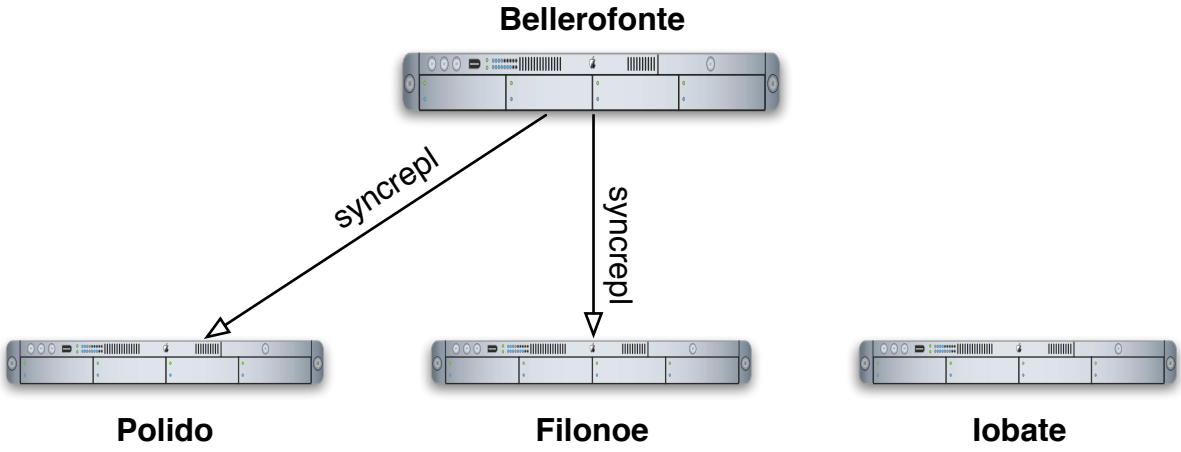
```
(1)   syncrepl rid=123
(2)    provider=ldap://10.4.4.1:389
(3)    type=refreshOnly
(4)    interval=0:00:00:10
(5)    searchbase="o=CeDoc"
(6)    filter="(ObjectClass=*)"
(7)    scope=sub
(8)    attrs=""
(9)    schemachecking=on
(10)   bindmethod=simple
(11)   binddn="cn=Manager,o=CeDoc"
(12)   credentials=secret
```

La sincronizzazione è basata su una query, quindi ne eredita molte caratteristiche. La riga (1) identifica il sistema come consumer e fornisce un identificatore di procedura di sincronizzazione. L'identificatore deve essere univoco. La seconda riga specifica l'URL del sistema provider da interrogare. La riga (3) specifica il tipo di sincronizzazione da effettuare. refreshOnly produce query di confronto a intervalli fissati, il risultato di ciascuna query propaga le informazioni di sincronizzazione. refreshAndPersist produce una singola query in una sessione persistente, ogni intervallo di tempo la query viene aggiornata propagando modifiche che differiscono nello stato della query. La riga (4) specifica l'intervallo di refresh della query. Le righe da (5) a (8) definiscono le caratteristiche della query: da che DN partire (searchbase),

che filtro utilizzare (filter, tipicamente si utilizza objectClass=* per ottenere tutte le entry indistintamente), che tipo di espansione si esegue sulla searchbase (scope) e a quali attributi limitare la sincronizzazione (attrs). La sincronizzazione preleverà tutti i risultati della query di ricerca e li confronterà con i corrispondenti presenti sul consumer per determinare cosa modificare. La riga (9) specifica se controllare che le entry modificate o aggiunte per la sincronizzazione rispettino gli schema del consumer. Infine, le righe da (10) a (12) specificano le modalità di autenticazione presso il provider: che tipo di autenticazione eseguire (in chiaro, con SASL o altro), che DN usare per l'accessso (binddn) e che password fornire (credentials).

Segue un esempio del posizionamento dei servizi di directory nell'infrastruttura attiva preso il CeDoc.

Figura 3.4: Struttura delle basi dati del progetto Bellerofonte



Il server LDAP principale è Bellerofonte. La sua funzione è recepire tutte le operazioni in scrittura e le interrogazioni generate dal servizio Proxy per l'accesso a internet. La macchina che ospita il server LDAP fornisce inoltre supporto alla struttura principale del Bus Java. Polido gestisce le interrogazioni provenienti da servizi accessori non gestiti direttamente dal CeDoc. Filonoe è utilizzato come basi dati di backup: può essere sostituito a Bellerofonte in caso di fallimento del sistema. Filonoe ospita inoltre una versione di backup dell'Interfaccia Web. Iobate è utilizzato soltanto per test e non viene sincronizzato con il resto del gruppo.

Le frecce (▷) indicano la direzione dei processi di replicazione e sono orientate da provider a consumer. Polido e Filonoe sono oggetto di replica e di nessun altro tipo di scrittura.

3.4.4. Altri parametri di configurazione

L'installazione predefinita di una directory OpenLDAP su sistema Unix utilizza i seguenti percorsi per la configurazione:

- /usr/local/etc/openldap: directory principale per la configurazione;
- /usr/local/etc/openldap/schema: directory per i file schema;
- /usr/local/etc/openldap/slapd.conf: file di configurazione principale della directory;
- /usr/local/etc/openldap/ldap.conf: file di configurazione per i client LDAP.

Il file slapd.conf contiene le direttive principali per la configurazione del servizio di directory, tra di esse sono presenti:

1. Specifiche sui percorsi assoluti di ciascun file schema da includere;
2. Argomenti a linea di comando e posizione del file descrittore di processo (pid file);
3. Moduli da caricare dinamicamente (ad esempio per la gestione del backend);
4. Regole di accesso (ACL);
5. Caratteristiche della base dati (formato del backend utilizzato, posizione sul filesystem, DN da usare come amministratore, sistema di autenticazione e password di amministrazione);
6. Specifiche degli indici;
7. Specifiche di sincronizzazione.

Segue un estratto di uno dei file utilizzati per la configurazione della base dati di test (Iobate):

```
# 1 - Inclusioni Schema
include /usr/local/etc/openldap/schema/core.schema
include /usr/local/etc/openldap/schema/cosine.schema
include /usr/local/etc/openldap/schema/inetorgperson.schema
include /usr/local/etc/openldap/schema/nis.schema
```

```
include /usr/local/etc/openldap/schema/openldap.schema
include /usr/local/etc/openldap/schema/cedoc.schema

# 2 - Argomenti e Pid
pidfile /var/run/openldap/slapd.pid
argsfile /var/run/openldap/slapd.args

# 3 - Moduli dimanici (per il backend Berkeley DB)
modulepath /usr/local/libexec/openldap
moduleload back_bdb

# 4 - ACL
access to dn.children="ou=biblioteche,o=CeDoc" attrs=userPassword
 by anonymous auth

access to dn.children="ou=utenti,o=CeDoc" attrs=uid, userPassword, CeDocMoDataUltimoCambio,
CeDocMoDataUltimaAttivazione, CeDocMoServizi, entry
 by dn.base="userid=proxy,ou=servizi,ou=CD,ou=biblioteche,o=CeDoc" read

# 5 - Specifiche della base dati
database bdb
suffix "o=CeDoc"
rootdn "cn=Manager,o=CeDoc"
rootpw secret
directory /home/ldap/data

# 6 - Indici
index objectclass,entryCSN,entryUUID                                    eq
index sn,givenName                                                             eq,pres,sub
index uid,userPassword                                                  eq,pres
index CeDocMoDataUltimoCambio,CeDocMoDataUltimaAttivazione eq,pres,sub
index CeDocMoServizi                                                    eq,pres
index CeDocMoSerialeScheda                                              eq,pres,sub
index CeDocMoIdDestinatario,CeDocMoIdOperatoreScheda      eq,pres
index CeDocMoNumeroTessera,CeDocMoProprietarioTessera           eq,pres

# 7 - Syncrepl consumer
syncrepl rid=123
 provider=ldap://10.4.4.1:389
 type=refreshOnly
 interval=0:00:00:10
 searchbase="o=CeDoc"
 filter="(ObjectClass=*)"
 scope=sub
 attrs=""
 schemachecking=on
 bindmethod=simple
 binddn="cn=Manager,o=CeDoc"
 credentials=secret
```

La versioni più recenti di OpenLDAP supportano un modello di configurazione in grado di inserire tutte le direttive di competenza del file slapd.conf nella directory stessa. La configurazione iniziale viene inserita a database vuoti attraverso un file LDIF[ii], ovvero il formato standard di specificazione dei dati usato da LDAP. Il file viene LDIF viene inserito nella directory la prima volta e ulteriori modifiche sono applicate direttamente sulla configurazione attiva e non richiedono il reload del servizio per essere attivate come configurazione corrente. Questo metodo implementativo è stato considerato per future modifiche al progetto Bellerofonte.

[ii] LDIF sta per LDAP Interchange Format ed è un formato utilizzato per descrivere entry LDAP in un file di testo. Spesso iil formato LDIF è utilizzato per produrre entry da inserire nella directory oppure per raccogliere direttive di modifica di entry esistenti.

4. ESTRAZIONE DEI DATI DALLE ANAGRAFICHE PREESISTENTI

4.1. Architettura del sistema di trasferimento dati

L'architettura intermedia di trasferimento dati all'interno del sistema è caratterizzata da tecnologie a supporto dell'interazione con i database legacy. I nuovi servizi basati sull'interazione con il database DLAP dispongono di un modulo adatto alla comunicazione diretta. Diversamente, per prelevare informazioni da basi dati legacy o, in generale, da sistemi che divengono "provider" di dati, è necessario adottare scelte tecnologiche differenti. Nella fattispecie si è scelto di realizzare un'interfaccia per ottenere le informazioni, trasformarla il accordo alla nuova logica applicativa e renderle accessibili attraverso le nuove tecnologie impiegate.

La metodologia alla base di questa scelta è simile a quella utilizzata per datawarehouse, ovvero strutture dati che riformulano informazioni presenti in altre strutture per adattarne le caratteristiche a interrogazioni di natura differente alle quali tali strutture fanno riferimento. In questo caso non si pretende di applicare logiche di aggregazione, altrimenti comuni a procedimenti di realizzazione di datawarehouse, poiché l'obiettvo primario è soltanto estrarre una porzione dei database legacy (le anagrafiche) per un utilizzo con OpenLDAP.

Si evidenzia che questo approccio, sebbene coinvolga più di una anagrafica, differisce in modo fondamentale dalle logiche più diffuse di integrazione delle informazioni. L'integrazione delle informazioni, intesa in senso accademico, non è caratterizzata dalla "copia" delle stesse. Spesso si tende a realizzare uno schema di rappresentazione comune (un'ontologia, per esempio) da utilizzare come vista delle sorgenti di informazione. In questo caso è stato ritenuto più opportuno un approccio caratterizzato da estrazione - conversione - copia, per favorire le prestazioni globali del sistema di ricerca e per arricchire le anagrafiche preesistenti di informazioni necessarie a gestire un processo di autenticazione.

4.1.1. Struttura del sistema di trasferimento

Il sistema di trasferimento è strutturalmente simile alla maggior parte dei middleware[i] enterprise che devono fare da collante tra backend di vario genere e logica applicativa. Queste infrastrutture sono spesso caratterizzate da un sistema di trasferimento e molteplici interfacce di interazione con i backend e con le applicazioni che sfruttano i dati.

Nel progetto l'infrastruttura ha la forma di un bus di trasferimento con due categorie di elementi che concorrono alla manipolazione dei dati:

- Producer: i componenti di tipo producer sono le interfacce con il backend. Il loro scopo è di rimanere attivi e di eseguire interrogazioni periodiche su sottoinsiemi dei dati presenti nei due database e veicolare i dati all'interno del bus in un formato comune;
- Cruncher: i moduli cruncher vengono attivati su richiesta all'arrivo di nuove informazioni dai producer. Tra di essi compaiono moduli di validazione dei dati e moduli in grado eseguire le modifiche al DIT LDAP necessarie a integrare le nuove informazioni. Questi moduli gestiscono una prima fase di rilevamento dei duplicati tra le entry e hanno un sistema di gestione degli errori di inserimento.

Si noti che il sistema a bus non supporta le letture arbitrarie sul DIT, svolte invece da moduli separati e specifici per le applicazioni che ne hanno necessità. La funzione del sistema di trasferimento è di veicolare i dati dai backend legacy al nuovo backend LDAP, applicato politiche di conversione e pulizia. L'interazione con il database LDAP è perciò limitata all'inserimento di dati attraverso le letture e le scritture necessarie.

L'architettura del bus consente una metodologia di lavoro completamente asincrona, ovvero garantisce la possibilità di moduli di operare in modo disgiunto l'uno dall'altro. Un modulo cruncher viene "svegliato" al sopraggiungere di una coda di messaggi provenienti da un producer e inizia ad elaborare le informazioni svuotando progressivamente la coda. Questo

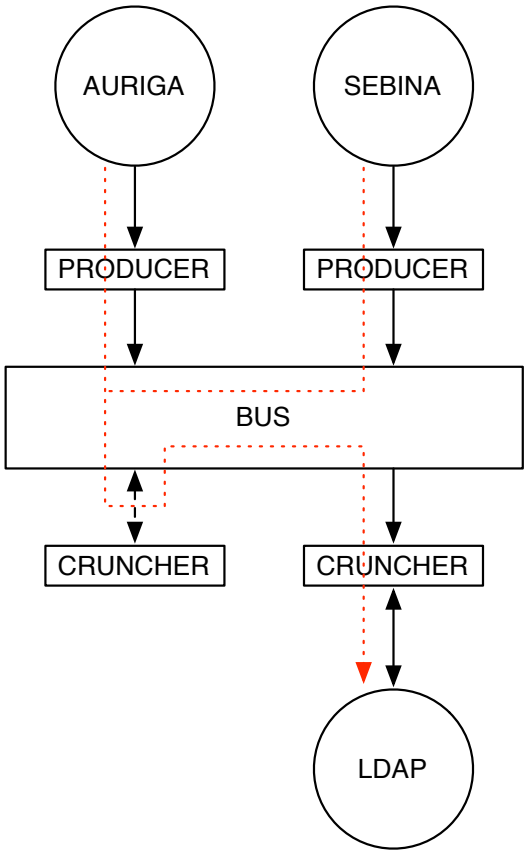
[i] Con il termine middleware si indica generalmente un software che unisce altri componenti sotware o applicazioni. Molto spesso architetture di middleware vengono utilizzate a supporto di applicazioni distribuite che includono server web, server applicativi e backend.

non impedisce al producer di sfruttare la coda come una pipe asincrona e continuare a fornire messaggi.

L'infrastruttura completa favorisce l'estrema distribuzione: il bus principale in cui viaggiano i messaggi può essere gestito su un nodo di una rete e i moduli possono operare su altri nodi sfruttando lo stesso stack di comunicazione che sfrutterebbero se si trovassero su di un solo nodo.

Nell'implementazione del progetto esiste un solo bus, al suo interno viaggiano messaggi trasferiti tra due differenti nodi producer, delegati all'interazione con Sebina e Auriga. I nodi producer elaborano la prima rappresentazione delle entry per il nuovo backend, la quale viene progressivamente raffinata in alcuni nodi cruncher fino a raggiungere lo stato definitivo in cui è pronta a essere scritta sul DIT. L'oggetto scambiato attraverso i messaggi prende il nome di Domain Object (DO), in accordo con la nomenclatura tipica dell'implementazione.

Figura 4.1: Struttura del sistema a bus



Il percorso in rosso indica la sequenza di moduli attraversati dal DO. Il modulo cruncher a sinistra ha il connettore tratteggiato a indicare la possibile presenza di più moduli dello stesso tipo.

4.1.2. Tecnologie a supporto del Bus

L'architettura del Bus realizzato per il progetto è coerente con le specifiche di quello che viene definito un Enterprise Service Bus (ESB). In generale, un ESB è un software o un'architettura implementata attraverso tecnologie di middleware e spesso basata su un paradigma a Web Service la quale fornisce i servizi fondamentali ad architetture orientate ai servizi di natura più complessa, attraverso un motore guidato da eventi e spesso basato su messaggi XML. Un ESB si propone come livello di astrazione calato al di sopra di un sistema di messaggistica, il quale consente agli sviluppatori di gestire il motore di trasferimento messaggi senza doverne scrivere il codice. La struttura tipica di un ESB predilige la decomposizione dei servizi in funzioni autonome, per favorire la distribuzione della tecnologia attraverso l'infrastruttura che fa uso del Bus. Questo è in contrasto con il noto paradigma a stack con funzioni gerarchiche accentrante in un unico provider di servizi all'interno di una rete.

Il progetto è stato realizzato sfruttando le tecnologie per la realizzazione di ESB messe a disposizione da sistemi basati sul linguaggio Java. In particolare, il sistema di messaggistica a supporto è JMS[ii]. La tecnologia JMS, parte integrante del panorama tecnologico di Java Enterprise Edition, definisce un set comune di concetti per la messaggistica su sistemi enterprise. Il suo scopo è minimizzare la quantità di concetti da utilizzare nel linguaggio Java per implementare un sistema di messaggistica e, nel contempo, migliorare la portabilità del sistema stesso su architetture distribuite.

JMS definisce alcuni concetti generali, tra cui:

- JMS Provider: entità che implementano JMS come sistema di messaggistica. Idealmente un JMS Provider è scritto interamente in Java, sfruttando molti dei vantaggi del linguag-

[ii] La Java Message Service (JMS) API è uno standard di messaggistica che consente ad applicazioni basata su Java di creare, inviare e ricevere messaggi. Il sistema consente la realizzazione di sistemi di comunicazione asincroni e lascamente accoppiati. Maggiori informazioni presso http://java.sun.com/products/jms/index.jsp

gio, tra cui l'essere eseguibile in applet e poter operare attraverso numerosi sistemi operativi e architetture;

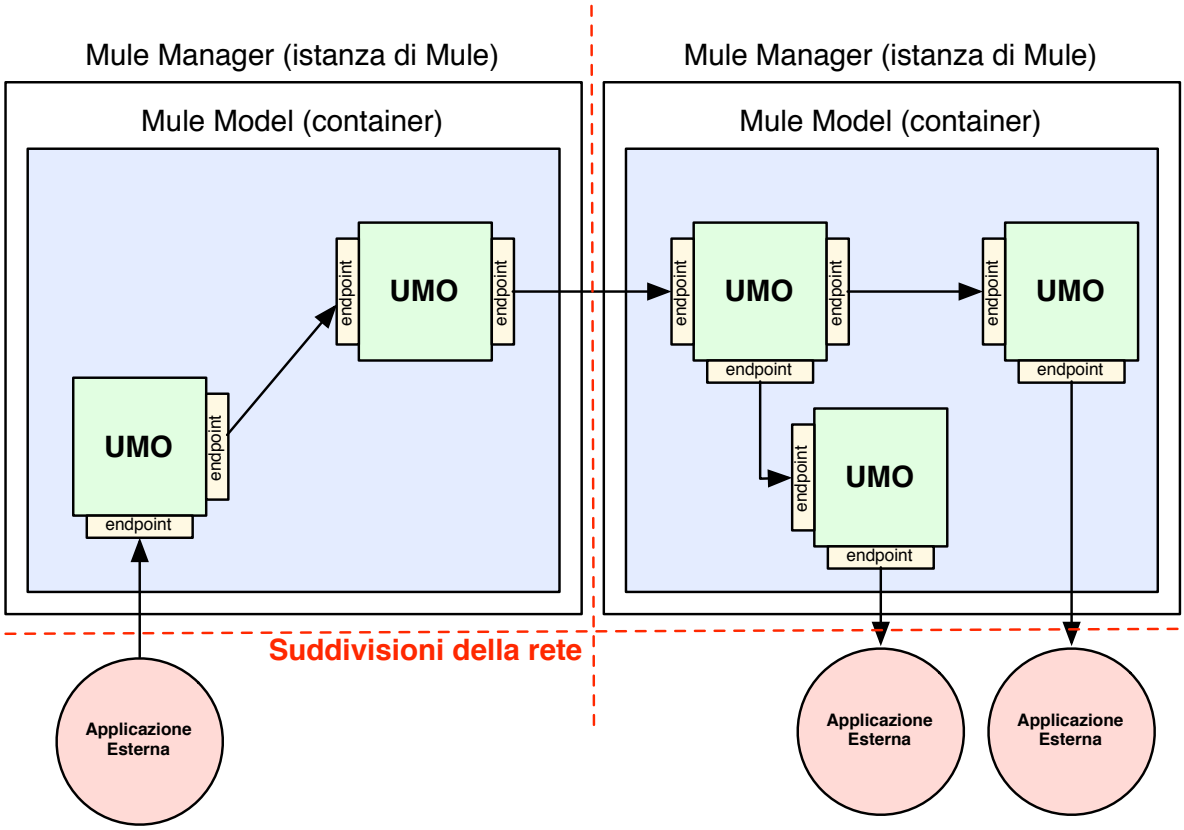
- JMS Message: interfacce per la generazione di messaggi. JMS offre un set di API da implementare nella realizzazione di Provider per la realizzazione di messaggi indipendenti dal provider stesso;
- JMS Domain: JMS si cura di definire la tipologia di delivery dei messaggi, offrendo strategie punto a punto o a pubblicazione e sottoscrizione (temporalmente disaccoppiati). Lo sviluppatore può utilizzare metodi già pronti per consegnare messaggi in entrambi i Domain (entrambi i paradigmi).

La tecnologia JMS è orientata all'estrema portabilità e consente numerose implementazioni. Nel progetto si fa uso di JMS come sottosistema a supporto del prodotto Open Source Mule[iii].

L'architettura di Mule offre strumenti configurabili, interamente realizzati in Java, per la composizione di un sistema basato su eventi che scambia messaggi in un formato derivato da XML. Il sistema di messaggistica a supporto implementa l'architettura necessaria per suddividere il concetto di bus in un insieme di elementi, ciascuno dei quali è una singola istanza di Mule. Il ruolo di questi elementi, detti Mule Managers è quello di far vivere al loro interno alcuni servizi a supporto dell'architettura e un container con più componenti di comunicazione. Il container dei servizi (detto Mule Model) offre il supporto a numerosi oggetti (detti UMO, o Universal Message Objects), ciascuno dei quali dispone di endpoint per la comunicazione (interfacce) e di una logica applicativa interna per elaborare le informazioni.

[iii] Sorgenti, binari e documentazione disponibili presso http://mule.codehaus.org/

Figura 4.2: Generica infrastruttura basata su Mule



Un'architettura basata su Mule con un buon livello di distribuzione utilizza degli UMO come interfacce di comunicazione con applicazioni esterne, le quali "parlano" con l'applicazione scambiando dati con protocolli e formati da essa dipendenti. Successivamente, le informazioni immesse dagli UMO terminali all'interno del Mule Model vengono scambiate tra un numero arbitrario di UMO che vivono solo all'interno del container. Ciascuno potrebbe ipoteticamente operare modifiche o validazioni all'informazione, prima di consegnarla all'oggetto successivo. Le informazioni possono essere veicolate anche tra più istanze di Mule (logicamente attive su più nodi della rete), utilizzando UMO che hanno endpoint di comunicazione tra istanze. Le informazioni possono uscire dal sistema di middleware attraverso altri UMO interfacciati con le applicazioni che necessitano del dato. Il sottosistema di comunicazione può utilizzare una grande varietà di protocolli.

L'architettura di Mule è stata sfruttata come riferimento per la realizzazione del bus. La componente nota come Bus Java è una singola istanza di Mule, la quale conosce il percorso definito per le informazioni, veicolate nella forma generica del Domain Object. Attraverso il passaggio per alcuni moduli nella forma di normali istanze di classi java il DO viene ottenuto, verificato e corredato di informazioni necessarie alla formulazione della struttura da scrivere nel DIT. In altre parole, Mule regge il percorso del DO attraverso i moduli (non nella forma di UMO), a partire dai producer, attraverso i cruncher, fino al DIT LDAP.

4.2. Procedure

Il sistema a regime mette in pratica alcune procedure formulate in sede di progettazione. Queste procedure rendono l'intera architettura in grado di trasferire le informazioni da una delle anagrafiche preesistenti al DIT LDAP, nel modo considerato soddisfacente dai requisiti di sistema.

4.2.1. Importazione

Il primo passaggio dell'ottenimento delle informazioni è l'importazione dalle basi dati precedenti. Questa procedura è delegata a due moduli producer in grado di leggere un tracciato prodotto rispettivamente da Auriga e Sebina. Non è stato possibile, né sarebbe stato conveniente, realizzare un modulo di interrogazione diretta delle basi legacy. Il sistema si basa invece sulla possibilità di accedere a un sottoinsieme delle informazioni prodotto arbitrariamente dalle basi dati legacy.

Sia Auriga che Sebina producono file e i worker associati attendono che i file siano prodotti in un percorso noto per leggerli e ricavare le informazioni in essi contenute. Ciascun producer sa come interpretare le informazioni presenti nel tipo di file prodotto dal sistema associato.

A livello di politiche di gestione, l'evento scatenante la produzione del file è il salvataggio dell'anagrafica di un utente attraverso la base legacy. Al momento del salvataggio l'anagrafica viene considerata come modificata ed è candidata per l'importazione. A questo punto

la procedura di generazione del file (detta genericamente transazione) differisce a seconda della base di partenza:

- Il software di Auriga è stato modificato aggiungendo una chiamata alla sequenza di salvataggio per produrre la transazione su un file. Ogni volta che viene eseguito un salvataggio gli stessi dati vengono scritti su file, questo vuol dire che salvataggi successivi comportano la generazione di nuovi files, virtualmente indipendenti l'uno dall'altro;
- Per Sebina è stata richiesta una procedura di "scarico periodico", ovvero una procedura eseguita a intervalli regolari che produce un insieme di risultati, tutti nello stesso file. I risultati sono un dump di tutte le anagrafiche modificate dall'inizio della giornata fino al momento in cui la procedura viene eseguita, un utente per ogni riga del file. La procedura di scarico è eseguita ogni due minuti, pertanto costringe il producer a eseguire più volte la scansione di utenti già inseriti.

La differenza fondamentale tra i due sistemi è nella quantità di informazioni prodotte. Sebbene le transazioni lette vengano sempre spostate per evitare la rilettura, è stato necessario sviluppare un sistema di rilevamento duplicati per evitare di popolare il DIT con entry mai utilizzate perché prodotte in modo automatico da riletture delle stesse transazioni. Questo fenomeno coinvolge particolarmente l'importazione da Sebina.

Le informazioni importate compongono il set di attributi noti per il Domain Object e vengono consegnate al primo cruncher dell'ordine per la prima verifica. La verifica consiste in una prova di coerenza dei dati con le politiche di gestione formulate in precedenza. In particolare si verifica la forma dell'attributo dedicato a nome e cognome e la presenza di valori per tutti gli altri attributi obbligatori. Se la transazione non soddisfa i criteri, viene ignorata, altrimenti viene proposta per il procedimento di rilevamento duplicati, svolto dal modulo di interazione con LDAP (ultimo cruncher).

4.2.2. Rilevamento dei duplicati

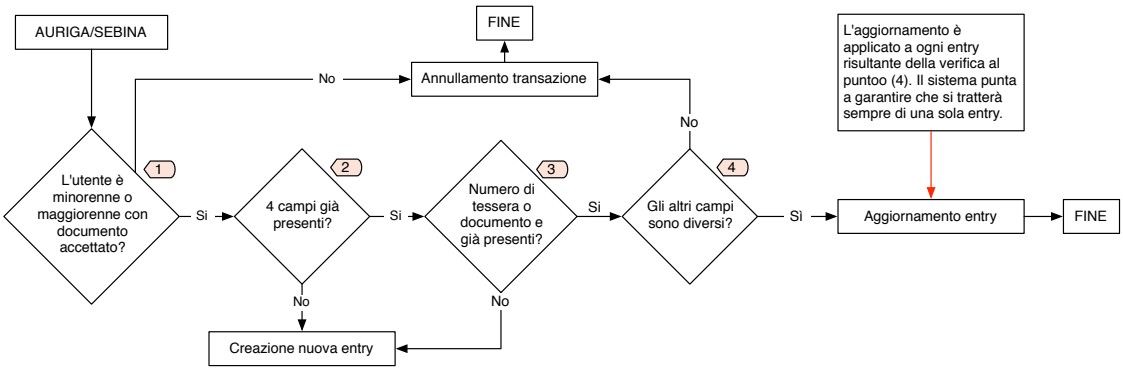
La procedura di rilevamento duplicati viene eseguita per ogni transazione che deve raggiungere il DIT e ha lo scopo di limitare il proliferare di entry duplicate all'interno del nuovo

backend. La verifica e identificazione dei duplicati consente inoltre di applicare modifiche e rettifiche a entry già presenti nel DIT, rispettando le specifiche procedurali che prevedono modifiche e inserimenti autorizzate soltanto a partire dalle basi precedenti.

Il primo punto di interesse riguarda la rappresentazione LDAP di una entry delle basi legacy. Quando un utente viene esportato dai database precedenti si ottiene una singola anagrafica contenente tutte le informazioni relative all'utente, compreso il numero di tessera. Questa anagrafica viene quasi interamente convertita nel gruppo di attributi formalizzati dalle classi CeDocMoPersona e CeDocMoAccount, tuttavia il numero di tessera bibliotecaria consente di generare un'altra entry, istanza di CeDocMoTessera. Ciò significa che l'ipotetico inserimento di una nuova anagrafica si traduce nella generazione di due entry LDAP.

Il seguente schema descrive il processo del primo livello di rilevamento, quello non interattivo e svolto all'interno del bus. Il secondo livello è eseguito attraverso la logica applicativa dell'interfaccia web ed è una procedura interattiva.

Figura 4.3: Primo livello di rilevamento duplicati



Il procedimento di rilevazioni attraversa le seguenti fasi.

1 - Verifica sul tipo di documento

Il sistema richiede che sia specificato un tipo di documento per consentire l'ingresso di una nuova entità. Tale verifica deve essere effettuata prima di tutte le altre e lascia passare soltanto utenti maggiorenni per cui è specificato un documento e utenti minorenni (che possono esserne sprovvisti). Questo controllo consente anche di escludere un'eventualità di confronto

non voluta: Auriga dispone di un valore fittizio per i documenti non specificati. Questo valore viene passato al sistema come viene inserito. Il risultato è che, utilizzando numero e tipo documento come discriminanti, si rischia di confrontare entry per cui non è specificato un documento e rilevarle come affini in base alla specifica dell'attributo fittizio, comune a tutte.

2 - Attributi identificativi

Si considera la sola entry per il ramo ou=utenti,o=CeDoc e si verifica una condizione di duplicazione su quattro campi identificativi dell'individuo: Nome, Cognome, data di nascita, sesso. Se la quaterna di attributi è già presente in almeno una entry del DIT, la nuova entry si considera un possibile duplicato e viene candidata per ulteriori verifiche. Il confronto sulla quaterna non esclude interamente la presenza di duplicati. Questa inefficienza è causata dall'impossibilità di utilizzare per la verifica anche il luogo di nascita, non fornito da Sebina. Ignorare il luogo di nascita rileva duplicati tra utenti nati lo stesso giorno e con lo stesso nome (e ragionevolmente lo stesso sesso), eventualità possibili e non certo rare. Considerare il luogo di nascita impone anche la località ed esclude gran parte dei duplicati di questo tipo, pur non essendo totalmente efficiente. Anche disponendo di cinque campi non è possibile distinguere i duplicati dai casi di comonimia (coincidenza totale dei cinque campi), presenti e gestiti con difficoltà anche da altri sistemi informativi. Per questo motivo vengono eseguiti ulteriori controlli prima di decretare un caso di duplicazione. Altrimenti, se i quattro campi non esistono in alcuna entry del DIT, la entry da inserire si considera nuova e si può procedere all'inserimento.

3 - Numero di tessera e numero di documento

Il secondo livello di verifica riguarda il numero di tessera bibliotecaria o il numero di documento. Le informazioni circa la tessera sono sempre presenti in ogni entry, a causa della loro provenienza. Informazioni circa il numero di documento non sono sempre disponibili. La verifica prevede di cercare tra le entry della query hit al punto (2) tutte quelle che hanno anche il numero di tessera e documento uguali. Se vengono trovate entry che soddisfano il requisito le informazioni sono sufficienti a decretare la condizione di duplicazione e a procedere con la verifica del punto (4). Se le entry positive al punto (2) non soddisfano il controllo sul numero di documento o numero di tessera, è possibile affermare che si tratta di utenti con

anagrafiche molto simili ma che fanno riferimento a documento o account diversi all'interno del sistema di prestito. Se un utente è stato iscritto due volte al prestito e ha due tessere diverse, probabilmente manterrà lo stesso numero di documento e verrà rilevato al controllo in questa fase. Se le due iscrizioni sono state effettuate con due documenti diversi non esiste altro modo per discriminare gli account e la procedura automatica è costretta a considerare due profili differenti. La risoluzione intelligente di questi duplicati sarà effettuata in modo interattivo in seguito.

Il confronto a questo passo è in grado di rilevare coincidenze tra anagrafiche differenti. Se un utente proviene da Auriga e scatena un duplicato per una anagrafica di Sebina, è possibile mantenere la condizione di duplicazione in base al numero di documento (se questo è uguale).

4 - Applicazione modifiche

Se una entry da inserire è stata identificata come un duplicato è possibile che si tratti di una rettifica degli attributi di un utente già presente. In tal caso la rettifica deve essere applicata alla entry. Dal momento che il caso di duplicazione è stato rilevato in base a coincidenze nei quattro campi identificati, appare evidente che una rettifica a monte di uno di questi campi scatenerà l'inserimento di una nuova entry nel DIT da gestire separatamente (e in seguito, in modo interattivo). Diversamente possono avvenire modifiche a informazioni accessorie nel profilo dell'utente. Le politiche di gestione evidenziano come unica modifica di scarso rilievo una variazione dell'indirizzo di domicilio. Se soltanto questo attributo è cambiato, la modifica può essere applicata alla corrispondente entry del DIT senza ulteriori variazioni. Se qualsiasi altro attributo è stato modificato è necessario impostare a 19700101 il valore dell'attributo CeDocMoDataUltimaAttivazione. Questo costringe gli operatori ad effettuare una ulteriore validazione dei dati dell'utente a seguito della modifica, prima che il profilo possa essere utilizzato nuovamente per l'accesso a internet.

Poiché le modifiche ai profili possono essere applicate da una qualunque delle postazioni di uno qualunque dei sistemi di prestito, esiste la possibilità che un operatore in una biblioteca modifichi un profilo al punto di richiedere la validazione dei dati e che tale validazione debba essere eseguita presso un'altra biblioteca in un momento futuro. Questo caso non

è un evento di rettifica, si tratta più probabilmente di una modifica accidentale. Il caso non costituisce un problema, bensì una condizione voluta.

Al punto (4) vengono scartate le entry identificate come duplicati che non apportano modifiche aggiuntive alle entry. Queste entry sono tipicamente dovute agli eventi di rilettura del file di dump da Sebina (frequenti e note) o a salvataggi successivi della stessa anagrafica in Auriga.

4.2.3. Scrittura sul DIT

La scrittura in sede di inserimento non presenta problemi di concorrenza rilevanti. Tuttavia si noti che le due scritture necessarie per realizzare la coppia Utente - Tessera non possono avvenire come una singola operazione, ovvero non è possibile formularle come un unico inserimento atomico. Quello evidenziato è un problema che coinvolge tutte le operazioni di modifica su LDAP ed è legato prevalentemente all'implementazione fatta del metodo di inserimento dalle librerie utilizzate. Questa caratteristica costringe l'applicativo che sfrutta le librerie a farsi carico del controllo di errori per garantire una forma di rollback delle modifiche eseguite prima del verificarsi dell'errore.

Per preparare la scrittura la entry viene generata come temporanea. Non è ancora disponibile un uid definitivo da utilizzare come nome utente per il collegamento: il nome sarà assegnato successivamente dall'operatore che per primo verificherà i dati dell'utente. La entry da inserire viene quindi generata con un uid temporaneo detto id di trans. L'uid viene generato in modo da garantire in ogni caso l'unicità all'interno del DIT. Un id di transazione è costituito dagli elementi:

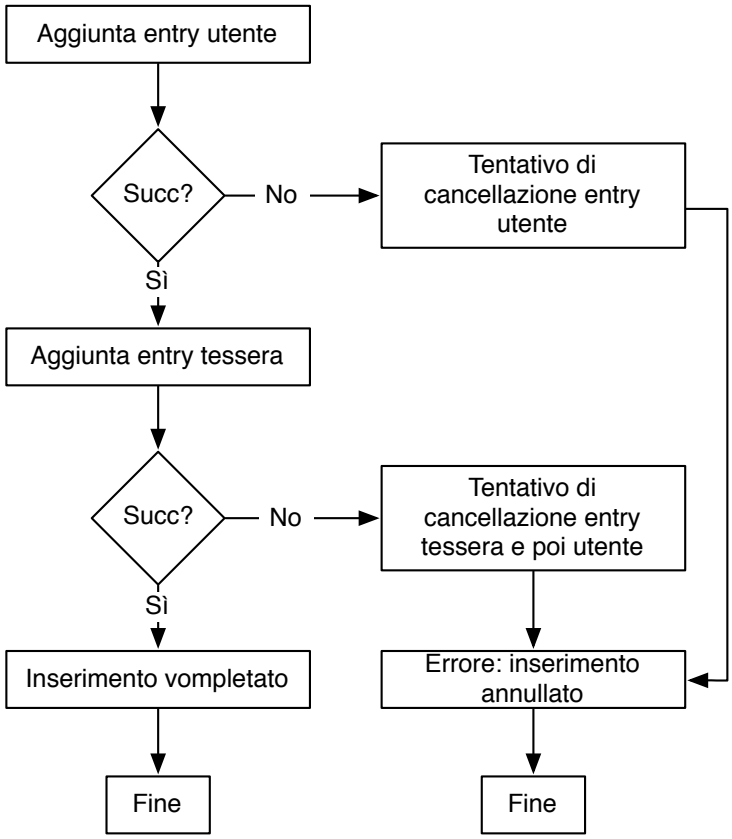
T-(data della transazione AAAAMMGG)(ora transazione ORE / MINUTI / SECONDI.MILLESIMI)

I caratteri in rosso sono parti fissate.

I restanti attributi del profilo utente e della entry per la tessera bibliotecaria sono completati in accordo ai valori noti e alle informazioni generate in automatico.

La scrittura sul DIT avviene in due fasi. La prima fase è la creazione della entry per l'utente. Il sistema cattura il risultato della prima fase e verifica se l'inserimento è andato a buon fine. Se l'inserimento ha uno stato di uscita diverso dal completo successo, l'operazione viene interrotta e si tenta di cancellare la entry inserita in modo non corretto. Se l'inserimento è andato a buon fine, si può procedere alla seconda fase: l'inserimento della tessera. Se l'inserimento va a buon fine l'operazione è conclusa, altrimenti è necessario il rollback completo della transazione, compresa la fase uno.

Figura 4.4: Procedura di inserimento



Il secondo tipo di scrittura previsto per questa fase è la modifica alle entry del ramo utenti (rettifica di attributi, punto 3). Un singola modifica di un numero arbitrario di attributi di un solo DN è vista dalle librerie di interazione con OpenLDAP come una singola operazione. Le modifiche alle entry utenti sono atomiche e non richiedono la gestione del rollback a livello applicativo.

5. PROGETTO E REALIZZAZIONE DELL'INTERFACCIA WEB

5.1. Requisiti

L'interfaccia operativa per gli utenti realizzata per il sistema è un punto di forte criticità. Per favorire l'accessibilità ai moduli per la raccolta di informazioni è stato scelto di realizzare un'interfaccia web. Un prodotto di questo tipo è caratterizzato da una logica di presentazione relativamente facile da realizzare e modificare e da una logica applicativa (middleware) di elevata complessità. All'interfaccia viene infatti delegata una porzione consistente delle interazioni con il backend LDAP, oltre a un fondamentale ruolo di validazione dei dati. Queste caratteristiche sono dettate dalla necessità di intervenire in modo mediato e controllato sul backend, costringendo gli utenti a operare in modo il più possibile allineato alle regole della base LDAP, traducendo i vincoli del database in caratteristiche della logica di presentazione.

5.1.1. Funzionalità

L'interfaccia deve svolgere la funzione di strumento per la gestione dei profili utente da parte del personale della biblioteca e deve fornire le funzionalità minime per la gestione dei dati rilevanti del profilo di ciascun utente, accessibili in modo immediato. Le funzioni principali possono essere elencate come:

1. Accesso con autenticazione: soltanto utenti che possano fornire un nome e una password corretti sono autorizzati a operare con l'interfaccia. Gli utenti non autenticati possono soltanto accedere a una pagina iniziale con contenuti informativi. L'autenticazione è seguita da una fase di autorizzazione, durante la quale l'utente riceve i privilegi di accesso relativi ai dati inseriti.
2. Ricerca degli utenti: il personale della biblioteca deve essere in grado di cercare utenti memorizzati nel backend senza dover interagire direttamente con esso. Si tratta di una caratteristica standard di prodotti web per l'interazione con database;

3. Accesso e modifica delle anagrafiche degli utenti: si tratta della funzione principale dell'interfaccia. La modifica delle anagrafiche deve essere possibile solo al personale delle biblioteca;
4. Assegnazione di schede con le password: registrazione del seriale di una scheda nel sistema per assegnare all'utente la password ad essa associata;
5. Rilevamento e rimozione guidata duplicati: non si tratta di un requisito esterno (ovvero utile agli operatori), si tratta di una necessità dettata dalla struttura del sistema di importazione. Il database richiede metodi di pulizia semiautomatici, eseguiti assieme agli operatori nell'ambito dell'interfaccia;
6. Modifica della password di accesso: la funzione deve essere accessibile a tutti gli utenti, i quali devono poter cambiare la propria password autonomamente, senza l'intervento del personale della biblioteca.

Le funzioni elencate sono caratteristiche della prima versione dell'interfaccia, proposta per l'avvio del servizio di autenticazione. Nuove estensioni sono attualmente in fase di sviluppo.

5.1.2. Utenti e ruoli

Gli utenti a cui si rivolgono le funzioni dell'interfaccia appartengono alle due categorie considerate nel resto del progetto:

- Utenti normali: tutti gli utenti sono utenti normali. Le persone che desiderano usufruire dei servizi informatici presso la biblioteca devono fornire i propri dati personali per l'inserimento nel sistema di autenticazione. Dopo aver fornito di dati ricevono un account nel sistema e dispongono di un nome utente e di una password. Per gli utenti normali l'unica procedura consentita è il cambio della password di accesso;
- Operatori: gli operatori sono utenti normali che in più possono gestire gli account del sistema. Agli operatori sono accessibili tutte le funzionalità elencate in precedenza. Per garantire a un utente i privilegi di operatore è necessario agire sulla sua scheda (funzione (3)) e attivare l'abilitazione. Gli operatori sono responsabili di tutte le operazioni che svolgono nel sistema ma non dell'operato degli utenti che abilitano.

Il livello di accesso come operatori è in genere garantito soltanto al personale delle biblioteca e al personale tecnico per la manutenzione del sistema. L'accesso come operatore richiede infatti addestramento alle funzioni del sistema . E' attualmente in sviluppo un nuovo livello di accesso per attivare funzioni amministrative di manutenzione, dedicato al solo personale tecnico di gestione. Il nuovo livello sarà supportato da nuove funzionalità dell'interfaccia web.

Le due tipologie di utenti descritte sono assimilabili a ruoli che gli utenti reali assumono all'interno del sistema. Per migliorare la progettazione dei percorsi di interazione è opportuno concentrare lo studio dei ruoli in base agli scopi che le categorie di utenti mirano a raggiungere attraverso l'utilizzo del sistema. Una volta focalizzati sugli scopi degli utenti è possibile progettare l'interfaccia che rende più agevole il raggiungimento di ciascuno scopo attraverso i percorsi oprativi disponibili.

L'unica funzione alla quale gli utenti normali possono accedere è il cambio password, pertanto lo scopo del profilo utente normale è quello di ottenere la modifica della password nel modo più veloce possibile, sia nei termini delle pagine da visitare sia nei termini della quantità di input da fornire al sistema. Data l'estrema semplicità, il requisito non interferisce con le politiche formulate in precedenza e può essere agevolmente soddisfatto.

Gli operatori interagiscono con l'interfaccia attraverso procedure più complesse. In base alle caratteristiche medie dell'ambiente di lavoro in cui il sistema viene utilizzato il prevalenza, è possibile effettuare alcune considerazioni sugli scopi del profilo utente degli operatori:

- Gli operatori operano spesso in condizioni in cui le procedure devono svolgersi il più rapidamente possibile. In caso di difficoltà a comprendere eventuali errori di inserimento o a reagire a eventuali errori di sistema, l'operatore può essere spazientito a causa del tempo che sta facendo attendere all'utente da abilitare. Lo scopo degli operatori è interagire con il nuovo sistema senza rallentamenti sulle altre procedure della biblioteca. Se questo fondamentale requisito non viene rispettato può crescere l'ostilità degli operatori verso l'interfaccia, vista come una inutile complicazione;

- La procedura da rispettare per l'inserimento degli utenti può essere difficile da apprendere per alcuni operatori. Il sistema deve tollerare tentativi ripetuti di invio delle informazioni e notificare chiaramente quando le operazioni sono andate a buon fine. Questo rende l'interfaccia più resistente a un utilizzo da parte di utenti inesperti. Obiettivo degli operatori è di svolgere le proprie mansioni senza dover avere cura di provocare soltanto interazioni corrette;
- Gli operatori effettuano attivazioni ma anche rettifiche dei dati degli utenti attraverso l'interfaccia. Lo scopo principale è riuscire in entrambe le procedure in base ai parametri precedentemente definiti. Queste procedure devono essere agevolate dall'interfaccia ma non una a scapito dell'altra. Ad esempio una procedura di inserimento guidata e suddivisa in passi è accettabile ma non deve essere applicata alla semplice rettifica, effettuata più efficacemente attraverso la valutazione di un modulo.

L'interfaccia web deve soddisfare gli obiettivi di ciascun profilo utente. E' abbastanza evidente che, data la natura ereditaria dei profili, sono necessari due differenti livelli di accesso e che una sola interfaccia non può soddisfare i requisiti di tutti e due i profili. Questa distinzione si applica prevalentemente alla necessità di autenticazione: gli operatori devono fornire un nome utente e una password per accedere a numerose procedure successive, gli utenti normali devono altresì certificare la loro identità, ma per accedere a una singola procedura separata dalle altre. Questa considerazione è sufficiente a individuare le caratteristiche della funzione di cambio password, la quale deve al contempo autenticare ed eseguire la propria funzione, per ridurre il numero di passaggi.

5.1.3. Linee guida per la progettazione

Unitamente allo studio delle funzioni, degli utenti e dei loro scopi, si riportano alcune linee guida da tenere in considerazione nella realizzazione del prodotto. Alcuni principi derivano da uno studio critico del panorama attuale di prodotti simili a quello da realizzare, altri derivano da considerazioni sullo stato dell'arte nella realizzazione di interfacce uomo macchina.

1. Modello di navigazione lineare: il modello di navigazione dell'interfaccia è sempre lineare attraverso percorsi svolti di pagina in pagina. La scelta di proseguire visualizza, se possibile, la pagina successiva. La scelta di regredire conduce allo stato consistente più vicino nel percorso di navigazione a ritroso. Ad esempio la ricerca produce un insieme di risultati, dai quali è possibile accedere alla scheda di ciascuno. Dalla scheda è possibile tornare soltanto alla ricerca, non alla pagina di risultati, che è temporanea. La chiusura del browser web interrompe in ogni caso la sessione ed equivale al comando di uscita. Non esiste alcuna forma di collegamenti incrociati tra le pagine che violino la linearità dell'interfaccia;
2. Information Hiding: gli utenti che operano con l'interfaccia devono accedere soltanto alle informazioni a loro utili. Informazioni circa i profili utente potenzialmente non corrette vengono visualizzate soltanto se l'interfaccia ne richiede rettifica. In nessun caso viene notificato all'utente se un'informazione non utile non è disponibile;
3. Strumenti autodescriventi: l'interfaccia dispone soltanto di strumenti autodescriventi, ovvero in grado di esplicare la propria funzione senza ricorso ad aiuti esterni. La stessa filosofia si rispecchia sui tasti per l'interazione, per i quali non sono state utiulizzate icone di alcun tipo, soltanto descrizioni testuali;
4. Accessibilità: l'interfaccia, privata delle caratteristiche stilistiche (come colori o immagini), deve rimanere accessibile e completamente funzionale;
5. Corretta notifica degli errori: i prodotti in circolazione sono spesso carenti quando si parla di notifica degli errori. In molti casi il software appare incolpare l'utente per eventi che in ultima analisi sono dovuti a carenze nella logica di gestione delle informazioni. Nel corso della progettazione è bene considerare che il software è soggetto a un certo numero di eccezioni non prevedibili e non causate dall'utente; quando queste eccezioni compromettono l'operatività del sistema, deve essere cura del software notificare il problema, ponendo come punto focale della notifica il mancato raggiungimento di un obiettivo preciso, non l'errore in sé. In ogni caso il software non dovrebbe imputare all'utente errori che derivano da una gestione non corretta di dati accettati in fase di inserimento.

5.2. Progetto

Questa sezione riporta la struttura dell'applicazione e le procedure fondamentali che esegue. Le caratteristiche del prodotto sono conformi alle politiche formulate per l'intero progetto Bellerofonte e ai requisiti espressi per l'Interfaccia Web.

5.2.1. Struttura

L'Interfaccia Web si presenta agli utenti come un sito in cui le procedure sono svolte navigando attraverso una o più pagine. L'architettura a pagine differenti è tipica dei prodotti di questo tipo basati sul web e si adatta bene all'implementazione attraverso i metodi scelti. Le pagine presentano alcune componenti comuni per dare coesione all'intero prodotto ma hanno funzioni differenti, ciascuna delegata a un passo delle procedure.

Start

Pagina iniziale. Si tratta di una pagina statica che accoglie gli utenti presso il sito web che ospita l'interfacca. Dalla pagina Start è possibile accedere al cambio password o a quella che è stata definita Area Riservata, dalla quale accedere alla validazione delle anagrafiche, solo per operatori.

CambioPass

Pagina per il cambio della password. Dal momento che agli utenti normali è consentito soltanto cambiare la propria password, appare inefficiente costringerli a una preventiva procedura di autenticazione per poi accedere a questa unica funzione che insiste proprio sui dati di autenticazione. Per questo motivo, la pagina CambioPass al contempo autentica gli utenti e consente loro di specificare una nuova password. La pagina richiede di inserire: nome utente (nome per il login), vecchia password, nuova password due volte. Alla conferma il sistema tenta il cambio della password e notifica eventuali errori rimanendo nella pagina, in caso di successo il sistema ritorna alla pagina Start notificando il successo. Dalla pagina CambioPass è possibile tornare alla pagina Start anche annullando l'operazione.

Login

Se un utente richiede di accedere all'area riservata, viene proposta un semplice pagina di login. Vengono richiesti nome utente e password per proseguire. La pagina Login verifica

anche che l'utente sia abilitato come operatore prima di garantire l'accesso. Da questa pagina è possibile tornare a Start annullando l'operazione di login. In caso di errore di autenticazione la pagina di Login permane.

Ricerca

La pagina di ricerca consente di cercare gli utenti alternativamente per nominativo ("Nome", "Cognome", "Cognome, Nome" o parti di essi) oppure per tessera bibliotecaria (comprensiva del prefisso della biblioteca). La pagina dispone di due funzioni: "avvia ricerca" e "torna all'inzio". Avviando la ricerca si accede alla pagina dei risultati (o direttamente alla pagina per le anagrafiche, se esiste solo un risultato). Tornando all'inizio si annulla la procedura di login e ci si porta su Start.

Risultato

La pagina viene visualizzata se la ricerca produce più di un risultato. Se la ricerca non produce alcun risultato si torna alla pagina Ricerca. Se la ricerca produce un solo risultato si passa alla pagina successiva. La pagina dei risultati dispone di una funzione per ritornare alla ricerca e elenca ciascuna anagrafica ottenuta. Per ogni anagrafica vengono riportati: Nome Completo (cn), Data di nascita, ultima tessera bibliotecaria nota. Per ogni anagrafica è presente una funzione di accesso, la quale conduce alla pagina Utente per l'anagrafica stessa.

Utente

Pagina di riepilogo delle anagrafiche degli utenti. La pagina viene richiamata con riferimento a un utente specifico e viene compilata automaticamente con tutti i dati noti. Gli utenti vivono all'interno del backend solo dopo essere stati importati correttamente dalle anagrafiche precedenti, perciò esistono sempre informazioni con cui compilare automaticamente la pagina. le informazioni fondamentali che possono essere oggetto di rettifica mediata dall'anagrafica preesistente non sono modificabili attraverso l'interfaccia web: la procedura richiede infatti di operare sull'anagrafica precedente. La pagina Utente offre la possibilità di salvare le modifiche oppure ritornare alla pagina Ricerca senza salvare. Se viene attivato il salvataggio e le modifiche vanno a buon fine, l'utente viene ricondotto alla pagina di ricerca con un messaggio di conferma. In caso di errore la pagina Utente permane con un messaggio di errore. Il concetto alla base di questo comportamento è consentire agli operatori di lasciare la pagina

solo se i dati che rappresenta sono corretti e completi, sia che li abbiano modificati (salvataggio andato a buon fine) oppure attraverso l'annullamento dell'operazione.

Duplicati

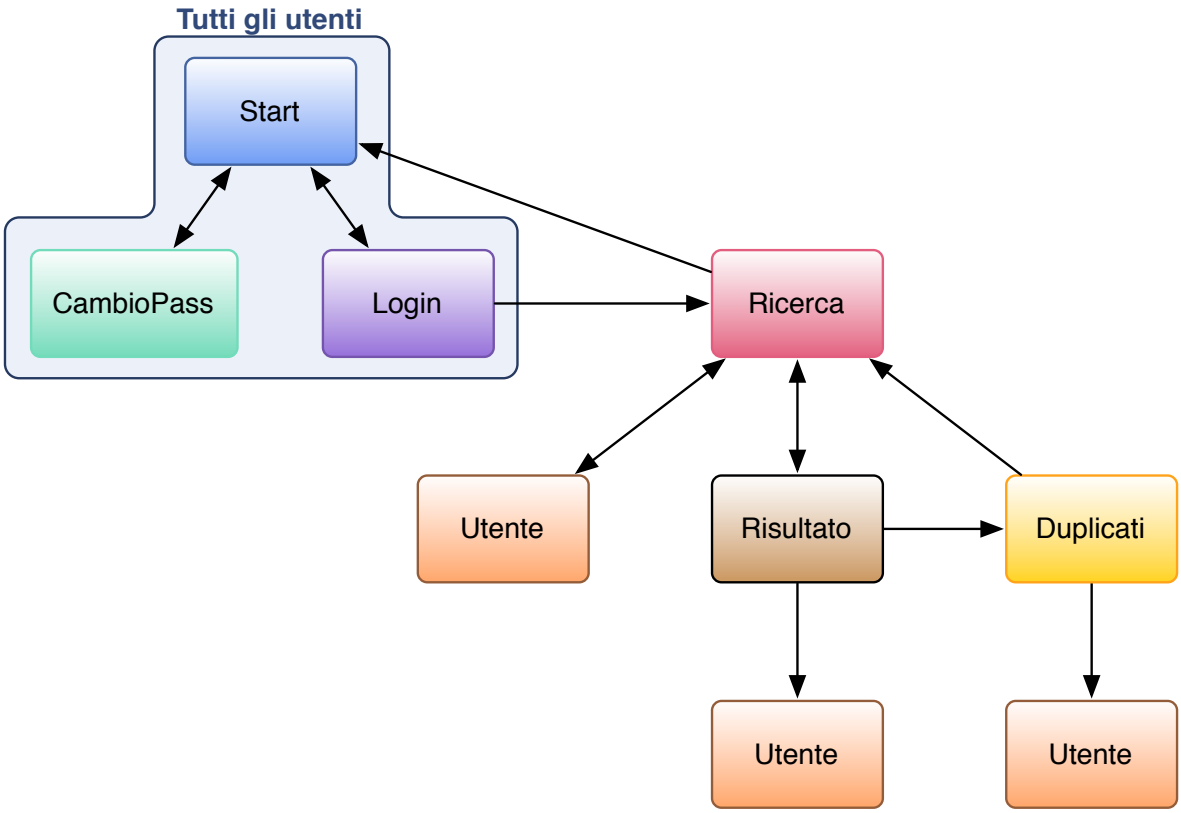
La pagina viene raggiunta in caso il risultato scelto dalla pagina Risultato soddisfi condizioni di duplicazione. La pagina Duplicati elenca tutte le anagrafiche per cui esistono condizioni di duplicazione. Per ciascuna anagrafica sono riportati: Nome, Cognome, Data di nascita, Sesso, Tipo di documento, Numero di documento, Tessera bibliotecaria. L'operatore può scegliere se tornare alla ricerca oppure se risolvere la duplicazione. La risoluzione comporta la scelta di un utente da modificare e comanda all'applicazione di impostare tutti gli altri presenti nella pagina come duplicati.

Errore

La pagina segnala un errore globale avvenuto nell'applicazione e propone la descrizione per gli errori più noti. Gli utenti possono tentare di tornare alla pagina precedente oppure riavviare la sessione di lavoro, ritornando alla pagina Start.

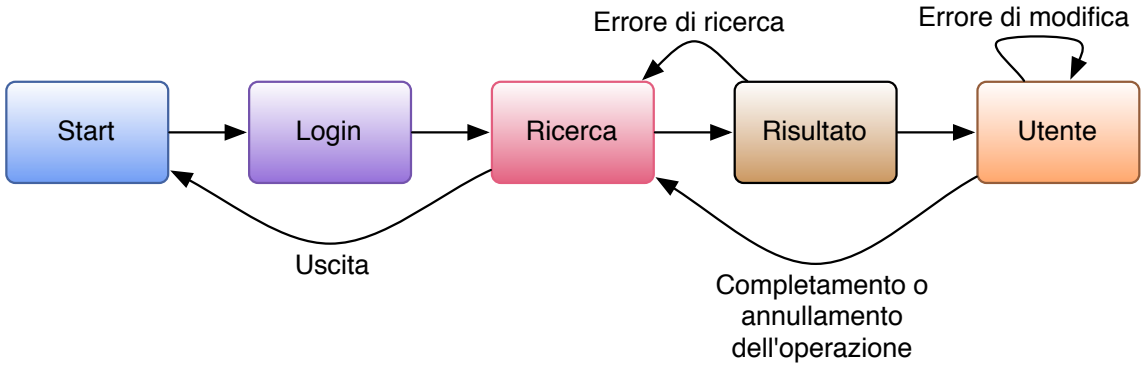
Nello schema seguente, le frecce indicano la possibilità di raggiungere la pagina che puntano partendo dalla pagina da cui partono. Le pagine sono riportate più volte per esaurire tutti i percorsi possibili nello sviluppo ad albero. La porzione in azzurro indica le pagine accessibili a tutti gli utenti, il resto dello sviluppo è accessibile soltanto agli operatori.

Figura 5.1: Sviluppo ad albero dei percorsi operativi dell'Interfaccia Web



La sequenza più comune di operazioni, considerando il sistema a regime, consiste in: Start -> Login -> Ricerca -> Risultato -> Utente. La sequenza modella l'accesso di un operatore che cerca un utente per abilitarlo al collegamento. L'operatore ottiene più di un risultato alla ricerca e ne sceglie uno da modificare.

Figura 5.2: Sequenza comune di operazioni



5.2.2. Procedura di ricerca

La logica dell'applicazione gestisce rappresentazioni a oggetti delle entità coinvolte nei processi gestionali. La categoria di entità che porta la maggior rilevanza informativa è evidentemente quella degli utenti. Per operare su entry di questo tipo il sistema gestisce i rami del DIT ou=utenti,o=CeDoc e ou=utenti-duplicati,o=CeDoc. In particolare il primo ramo viene gestito per le interrogazioni sulle entità di cui gestire i parametri, il secondo è la destinazione di entità spostate a seguito di quello che è stata definit0 secondo livello del processo di rilevamento dei duplicati. L'applicazione elabora inoltre informazioni di competenza ai rami ou=tessere,o=CeDoc e ou=schede,o=CeDoc. Da questi rami sono ottenute le informazioni necessarie al rilevamento dei duplicati e all'assegnazione di password agli utenti. Gli operatori percepiscono l'esistenza di entità distinte soltanto per quanto riguarda gli utenti, le restanti informazioni sono da intendersi a supporto della logica applicativa.

Esistono due procedure che vengono eseguite in modo implicito dall'applicazione in concomitanza con la ricerca e selezione degli utenti da modificare: rimozione di tessere "orfane" e rilevamento dei duplicati per risoluzione interattiva (descritta nella prossima sezione). Oltre a queste sarà trattata anche la generica sequenza delle operazioni in ricerca.

Ricerca

L'evento di ricerca è sempre attivato su richiesta dell'utente. La pagina Ricerca consente di cercare alternativamente per Nome e Cognome oppure per Tessera Bibliotecaria. Se gli operatori specificano lemmi per entrambi i campi soltanto il primo viene utilizzato. L'unico lemma di ricerca consente di formulare una query LDAP sul ramo ou=utenti,o=CeDoc alla quale viene richiesto di fornire tutti gli attributi noti per ciascuna entry trovata. Ogni entry ottenuta fornisce le informazioni per istanziare un oggetto nella logica applicativa, in questo modo le informazioni ottenute vengono incapsulate e mantenute nel corso della sessione, attraverso la navigazione per la pagina Risultato. Per ciascuna entry viene ottenuta la corrispondente entry del ramo ou=tessere,o=CeDoc. Se la tessera non esiste la entry non può essere visualizzata.

Nel momento in cui un operatore sceglie un utente da modificare vengono attivate le procedure di Rimozione di tessere orfane e Rilevamento duplicati.

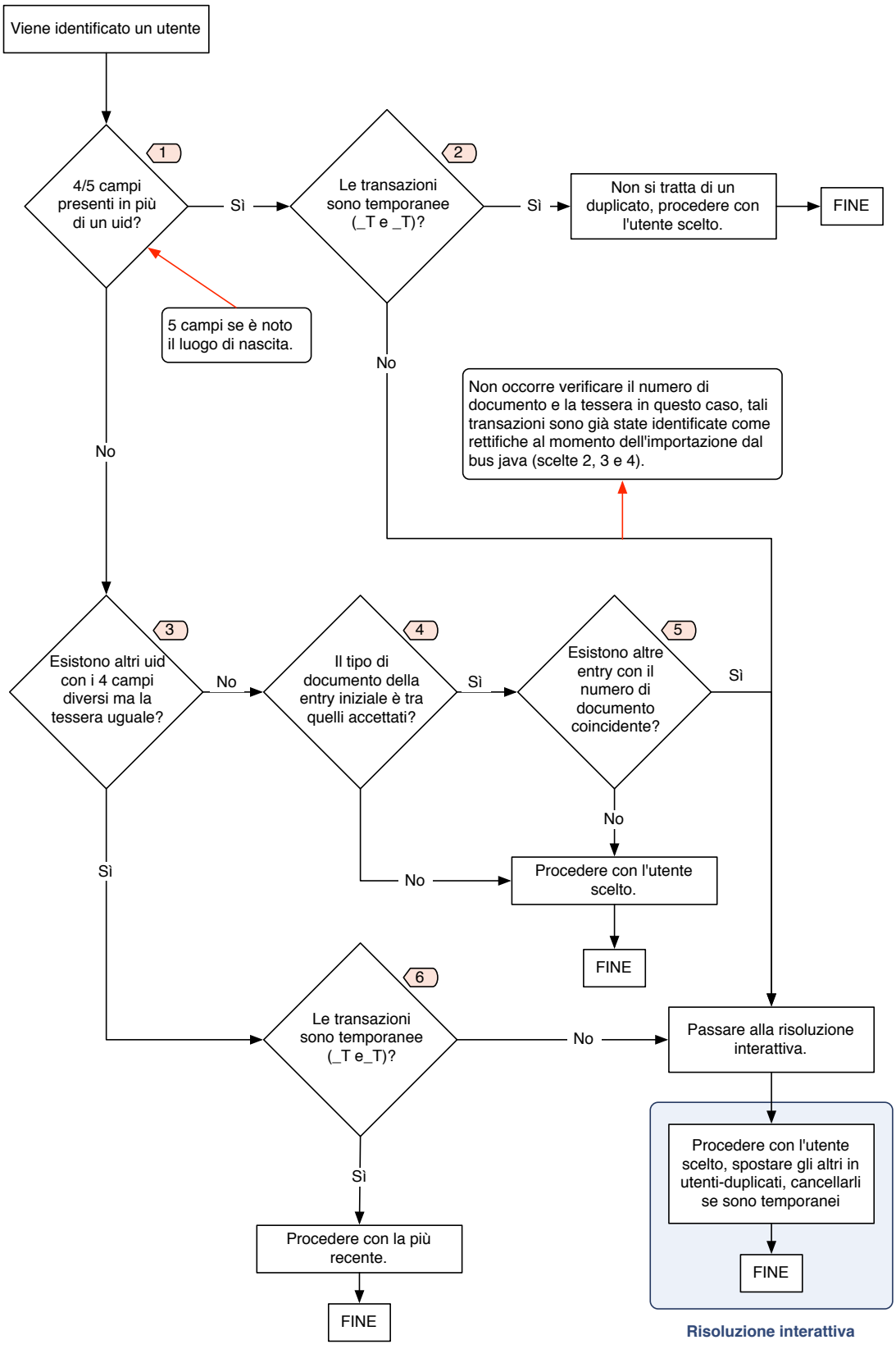
Rimozione di associazioni orfane

Nel DIT possono esistere entry del ramo ou=tessere,o=CeDoc che modellano tessere bibliotecarie in cui uno dei valori per l'attributo CeDocMoProprietarioTessera è nullo o fa riferimento a uid inesistenti. Le associazioni di tessere a utenti non esistenti, evidenziate nel corso della rilevazione dei duplicati, sono definite associazioni orfane. I valori che non hanno riscontro nel ramo ou=utenti,o=CeDoc possono essere rimossi senza compromettere il funzionamento del sistema; la rimozione è effettuata automaticamente man mano che le associazioni orfane vengono rilevate. Se accade che una tessera gadi sole associazioni orfane viene scatenato un errore. Questo evento non è considerato possibile nel contesto dell'operatività corretta del sistema, pertanto non viene gestito come anomalia in modo automatico.

5.2.3. Procedura di rilevamento duplicati

Il rilevamento dei duplicati delegato alla logica applicativa dell'Interfaccia Web deve garantire la massima precisione possibile, considerati i database di partenza. In caso di ambiguità non risolvibile in modo automatico, il sistema richiede l'intervento dell'operatore attraverso la risoluzione interattiva. Lo scopo della procedura di rilevamento è quindi quello di proporre una lista per la risoluzione da parte dell'operatore. La procedura di rilevamento dei duplicati è attivata ad ogni richiesta di modifica dei dati di un utente e prende il nome di secondo livello di rilevamento.

Figura 5.3: Secondo livello di rilevamento dei duplicati



Nei confronti si fa riferimento a coppie di entry o entry, in realtà possono esistere anche più di due entry alla volta sulle quali sussistono condizioni di duplicazione.

1 - Verifica dei campi identificativi

Ogni volta che viene richiesta la scheda di un utente, l'applicazione verifica se esistono altri utenti che hanno gli stessi 4 o 5 campi identificativi. All'interno dell'Interfaccia Web vengono gestite entry per cui può essere noto il luogo di nascita (temporanee provenienti da Auriga e attive), pertanto se possibile si fa uso del confronto sul luogo di nascita. In caso di rilevamento di duplicati in questo senso, si procede alla verifica dello stato delle entry. Diversamente su procede con la verifica sul numero di tessera e documento.

2 - Verifica dello stato delle entry

Se esistono entry per le quali sussiste una condizione di duplicazione per i 4 o 5 campi identificativi si impone un controllo sullo stato. L'Interfaccia Web opera sia con entry temporanee che con entry Attive. Se tutte le entry sono temporanee non esiste un reale problema di duplicazione, in quanto si tratta di profili non utilizzabili. Nel caso, è sufficiente procedere con la visualizzazione della entry richiesta dall'operatore e ignorare le altre. Diversamente, si è rilevato un caso di possibile duplicazione (senza dimenticare le comonimie) che deve essere risolto con l'aiuto dell'operatore. Questa eventualità (uscita "No" dalla condizione (2)) si verifica molto raramente tra entry attive. Generalmente sussiste una condizione di duplicazione tra entry attive e entry temporanee.

Nella fase (2) non occorre applicare verifiche su numero di tessera e documento. Tali verifiche sarebbero ridondanti rispetto a quelle eseguite nelle fasi precedenti nel bus (1, 2 e 3).

3 - Verifica del numero di tessera

Questa verifica è effettuata efficacemente ottenendo la tessera associata all'utente, poi dalla tessera l'elenco dei proprietari. Ogni proprietario diverso dall'utente stesso è un potenziale duplicato. Nella fase (3) giungono entry per le quali non è stata rilevata una duplicazione sui parametri identificativi di una persona (4 5 campi) e alle quali si applica una verifica su un campo relativo all'account (il numero di tessera). Se la risposta alla verifica è positiva, si

impone una nuova verifica sul documento. Se la risposta è negativa si applica l'ultimo controllo di duplicazione.

4 - Verifica del tipo di documento

Similmente alla verifica (1) effettuata dal bus, anche presso l'interfaccia devono essere scartate entry che dispongono di documenti non accettati. La verifica (4) scarta tutti gli utenti maggiorenni che non hanno un documento accettato e tutti i minorenni senza documento. L'implementazione di questa verifica è giustificata nel controllo successivo, ove il tipo e il numero di documento sono discriminanti. Si desidera applicare questo tipo di confronto soltanto a utenti per cui il valore è utilizzabile a tal fine. Se non è possibile applicare discriminazione, si procedere con successo, altrimenti si utilizza il documento per un ulteriore confronto.

5 - Verifica del numero di documento

Ultima verifica di duplicazione, numero di documento. Se due entry hanno i 4/5 campi diversi, la tessera diversa e il numero di documento diverso, sono entry indipendenti. Se la verifica (5) dà esito negativo si può procedere con la gestione dell'utente prescelto. Si tratta del caso più comune di uscita. Se la verifica ha esito positivo esistono più utenti memorizzati con lo stesso numero di documento, questo impone la risoluzione interattiva.

6 - Verifica dello stato delle entry

Due entry che hanno i 4/5 campi con differenze ma lo stesso numero di tessera possono ricadere nelle conseguenze delle riassegnazioni di account in Auriga trattate in precedenza, oppure possono essere salvataggi immediatamente successivi separati da lievi modifiche. E' possibile distinguere i due casi tramite lo stato della entry. Se le entry sono temporanee, si tratta certamente di salvataggi successivi, in tal caso è utile soltanto la più recente: si procedere con quella e le altre vengono cancellate. Se almeno una delle entry non è temporanea, può trattarsi di un caso di riassegnazione o di semplice rettifica. In tal caso è richiesto l'intervento dell'operatore per la risoluzione interattiva.

Risoluzione interattiva

La risoluzione interattiva è semplicemente una richiesta all'operatore della scelta di un utente. Tutte le entry utente per le quali è stata rilevata una condizione di duplicazione ai punti

(2), (5), o (6) vengono elencate nella pagina Duplicati. Per ciascuna entry vengono fornite informazioni sufficienti alla distinzione per favorire l'operatore. Ogni entry è associata a un bottone etichettato "Conserva solo questo utente". Selezionando il bottone tutte le altre entry subiscono lo spostamento se sono attive (si portano in stato Duplicato) e la cancellazione se sono temporanee (si portano in stato Eliminato). L'applicazione richiama poi la pagina Utente per la entry favorita dalla selezione. Lo spostamento o eliminazione delle entry sfavorite è accompagnato anche dalla rimozione di eventuali entry tessera associate (o di singoli valori dall'attributo CeDocMoProprietario).

5.3. Architettura

Sono disponibili numerose tecnologie per la realizzazione di prodotti web dinamici. Per omogeneità con il resto del software realizzato per il progetto, si è scelto di utilizzare strumenti basate su Java. La scelta è giustificata dalla necessità di realizzare uno strumento con una logica di presentazione relativamente semplice (poche pagine web) ma una logica interna di elevata complessità, in grado di gestire entry LDAP e numerose procedure di verifica.

5.3.1. Jakarta Tapestry

Le soluzioni basata su Java in genere fanno uso di un server web scritto nel medesimo linguaggio per il deployment di strumenti detti web applications. Le webapp si presentano come archivi non dissimili dai tradizionali jar (bundle per applicazioni Java), contengono una parte scritta in Java per la business logic e possono contenere una parte di interfaccia in grado di sorreggere un sito web.

L'interfaccia web è interamente realizzata utilizzando il framework component-based per la realizzazione di applicazioni web Jakarta Tapestry[i]. Tapestry non è un sistema di template in senso stretto; è invece utilizzato per realizzare siti web dinamici e interattivi appoggiandosi sulle Java Servlet API[ii]. L'approccio proposto è quello di costruire applicazioni web,

[i] Sorgenti e documentazione disponibili presso http://tapestry.apache.org/

[ii] Le Java Servlet API forniscono agli sviluppatori web un meccanismo semplice e consistente per estendere le funzionalità di un web server. Una servlet è sostanzialmente una applet in esecuzione presso il web server senza interfaccia utente. Maggiori informazioni presso http://java.sun.com/products/servlet/

anche ricche e complesse, attraverso la composizione di componenti riutilizzabili. Il framework gestisce internamente molte procedure come la manipolazione della sessione e delle richieste, l'internazionalizzazione, la codifica degli URL, riducendo sensibilmente il lavoro di programmazione a basso livello.

Gli sviluppatori si focalizzano sullo sviluppo di oggetti, metodi degli oggetti e proprietà in accordo con la convenzione JavaBeans[iii]. In un'applicazione web costruita con Tapestry l'azione dell'utente si traduce in una modifica delle proprietà degli oggetti collegati agli strumenti utilizzati dall'utente per esprimere l'azione. Tale collegamento è dichiarato dal programmatore e gestito interamente da Tapestry. Per ottenere l'associazione, il programmatore non scrive la servlet stessa, scrive un listener method; in altre parole è sufficiente scegliere un componente e configurare un listener per il componente che invochi un metodo, il quale viene esteso con qualsiasi funzione necessaria alla logica applicativa.

Queste caratteristiche rendono Tapestry un prodotto component-centric, diverso da molti strumenti per il web dinamico detti operation-centric. La differenza principale è nel modo di gestire le procedure nascoste dietro all'interfaccia, compito che Tapestry affida alle proprietà degli oggetti che il programmatore usa come componenti. Diversamente, il programmatore dovrebbe preoccuparsi di scrivere la logica applicativa e di connettervi le pagine in output manualmente.

La logica di Tapestry considera le pagine web dei componenti che a loro volta sono costituite da altri componenti, con vari livelli di annidamento. Quando una pagina viene richiesta, il motore di rendering ne ottiene il template, generalmente formulato in HTML o XHTML[iv], dal template è possibile distinguere gli oggetti di markup che hanno proprietà come componenti di tapestry ed eseguire il rendering di ogni componente in base a direttive definite o derivate dallo stato di oggetti nell'applicazione. Molte proprietà possono essere identi-

---

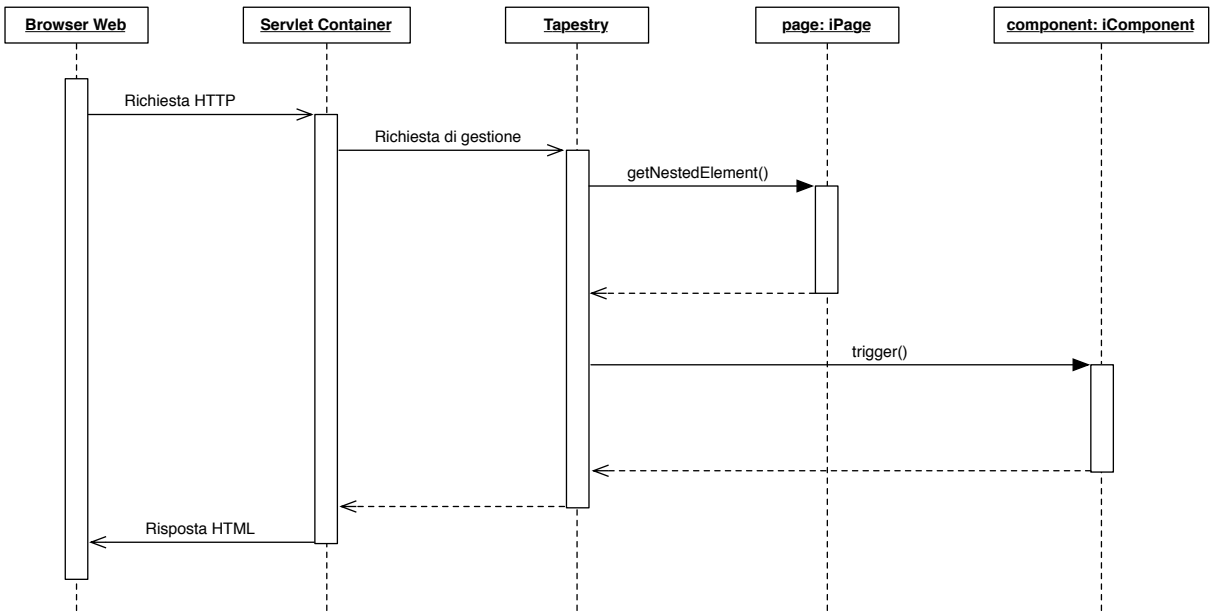
[iii] La tecnologia JavaBeans fa parte del'architettura Java 2 Standard Edition. Gli oggetti compatibili con lo standard JavaBeans possono essere riutilizzati attraverso numerose applicazioni e in generale assemblati con componenti esterne incluse nel progetto. Maggiori informazioni presso http://java.sun.com/products/javabeans/index.jsp

[iv] Rispettivamente HyperText markup Language e Extensible HyperText markup Language. Due linguaggi di markup standard per la realizzazione di pagine web. Gli standard sono mantenuti presso il World Wide Web Consortium http://www.w3.org/MarkUp/

ficate all'interno dell'applicazione facendo uso del linguaggio OGNL (Object Graph Navigation Language). OGNL è un linguaggio che utilizza espressioni Java per indagare all'interno di oggetti. La sua sintassi consente di identificare una proprietà di un oggetto in modo univoco specificando una sorta di percorso operativo. Le proprietà da ricercare all'interno di oggetti della logica applicativa possono essere efficientemente indicate via espressioni OGNL.

Il risultato del processo di rendering è una pagina web statica con numerose componenti attive come JavaScript per la validazione lato client. Le componenti attive vengono generate e gestite interamente dal framework.

Figura 5.4: Ciclo di una richiesta a un servlet container che fa uso di Tapestry



Il framework mette a disposizione numerosi componenti già pronti che il programmatore deve solamente configurare e collegare a oggetti nel linguaggio di markup. Inoltre è possibile definire componenti personalizzati e utilizzare elementi che non hanno lo scopo di produrre codice HTML ma elaborano informazioni internamente.

In base alla terminologia formulata dal gruppo di Jakarta Tapestry, ogni applicazione realizzata con il framework fa uso di alcuni componenti principali:

- Engine: è l'oggetto centrale di un'applicazione. l'Engine è per Tapestry quello che la HttpSession è per le Java Servlet API, in altre parole un Engine è responsabile di conservare e gestire la sessione dell'applicazione;
- Engine services: si tratta del ponte tra le servlet, gli URL e il resto di Tapestry. Gli Engine services sono responsabili di compiti come la codifica degli URL e mappare le azioni specifiche da generare in caso di attivazione di un determinato URL. I services gestiscono inoltre le richieste in arrivo e l'incapsulamento delle informazioni come proprietà di oggetti (per la tipica gestione fatta da Java);
- Visit Object: questo oggetto ha lo scopo di gestire lo stato del server. La classe che modella questo oggetto deve essere definita nell'ambito dell'applicazione ed è una proprietà per l'Engine;
- Global Object: un altro oggetto specifico per l'applicazione. Il Global Object è utilizzato per centralizzare la logica di lookup JNDI[v].

5.3.2. Componenti, classi, pagine e servizi

L'architettura dell'applicazione rispetta la struttura a pagine formulata in progettazione e la riporta nella logica di Jakarta Tapestry. Ogni pagina dell'interfaccia è una pagina per Tapestry, ovvero un particolare tipo di componente che verrà renderizzato in una pagina web completa. All'interno delle pagine sono utilizzati quasi solamente componenti standard di tapestry (come etichette o campi di input), i quali sono configurati per interagire con gli oggetti dell'interfaccia. L'eccezione a questa struttura è un componente "custom" ovvero realizzato su misura per l'applicazione chiamato Border. Lo scopo di border è di facilitare lo sviluppo delle pagine integrando le parti condivise da tutte come la grafica di contorno e il sistema unificato di segnalazione degli errori.

All'interno della struttura delle classi è stata realizzata la classe DPSBasePage, la quale estende la classe BasePage di Tapestry per la realizzazione di componenti pagina. Tutte le classi che estendono DPSBasePage sono pagine ed ereditano tutti i parametri configurati nella

[v] La Java Naming and Directory Interface è parte della piattaforma Java. Essa fornisce ad applicazioni basate su Java un'interfaccia unificata per molteplici servizi di naming e directory. Maggiori informazioni presso http://java.sun.com/products/jndi/

classe, utilizzati per fornire una configurazione predefinita a ciascuna nuova pagina inserita nel progetto. Le classi che modellano il comportamento di ogni pagina portano il nome delle pagine definito in progettazione. Il class diagram completo dell'interfaccia web è disponibile in appendice.

Un'altra classe ad uso globale dell'applicazione è DPSDelegate. Questa classe estende la classe ValidationDelegate standard di Tapestry e fornisce il servizio di cattura degli errori di validazione. Ogni volta che la validazione lato server di un'informazione genera un errore di qualche tipo, l'errore viene catturato dal Delegate che fornisce il servizio di interpretazione e notifica della causa. La personalizzazione del ValidationDelegate è essenziale per rendere la notifica degli errori omogenea con il resto del sistema. Gli errori che non riguardano la validazione ma si configurano come errori generici di sistema vengo ridiretti su una pagina di errore generica, la quale fornisce le due opzioni "Ritenta" e "Riavvia la sessione". Questa funzione è personalizzata e sostituisce la tipica notifica di tapestry con la pagina Errore. La classe delegata alla presentazione degli errori definiti "runtime exceptions" è DPSErrorPresenter.

Tra le classi globali sono presenti DPSGlobal e DPSVisit che realizzano rispettivamente il Global Object e il Visit Object di Tapestry. La seconda è più importante poiché il Visit Object deve mantenere in sessione tutte le caratteristiche dell'utente collegato, inclusi i parametri ottenuti attraverso la lettura del suo profilo LDAP.

Il procedimento di validazione lato server delle informazioni viene eseguito attraverso alcuni custom validator. Ciascuna classe per un validator estende la classe predefinita di Tapestry per i validator: BaseValidator.

Il sistema utilizza i seguenti elementi:

- Envelope: verifiche relative alla busta (o scheda per password) associata a ogni profilo utente;
- FineSosp: validazione della data di fine sospensione impostata in caso di sospensione dell'utente. La validazione riguarda sintassi e verifica temporale;
- Garante: verifica di presenza di un garante (genitore) per gli utenti minorenni. Il validator verifica che sia specificato un garante e che l'utente esista realmente;

- Residenza: verifica di presenza dei dati di residenza;
- TipoDocumento: verifica della specifica di un tipo di documento.

Infine sono disponibili tutte le interfacce utilizzate per la gestione delle transazioni a basso livello. Le pagine di Tapestry non includono la logica di interazione con basi dati o di verifica. Queste operazioni sono delegate rispettivamente a servizi e validator. I servizi sono funzionalità registrate presso un provider di servizi. La classe che modella ogni pagina può "iniettare" a tempo di esecuzione il codice dei servizi nel suo codice principale. I metodi messi a disposizione dai servizi, anch'essi scritti in Java, sono importati attraverso l'inclusione dell'interfaccia implementata dalle classi dei servizi. In altre parole, il sistema dispone di alcune interfacce e le relative classi che vengono gestite come servizi importati. Tapestry utilizza Jakarta Hivemind[vi] per istanziare servizi esterni a tempo di esecuzione. Hivemind fornisce la logica per la gestione dei provider di servizi, in questo caso compresi nel progetto. Le interfacce utilizzate sono:

- ILogin: verifica le credenziali fornite, è utilizzata da tutte le pagine in cui viene attivato un processo di autenticazione (Login e CambioPass);
- IRetrievalInfo: utilizza il modello utente realizzato per il progetto Bellerofonte per fornire un oggetto che modella l'intero profilo utente. Utilizzata in ogni situazione in cui è necessario gestire profili utente;
- IUtenteModifications: gestisce l'intero apparato di scritture LDAP effettuate dall'interfaccia web.

Le interfacce forniscono la traccia per l'implementazione delle funzioni di ciascun servizio, nominate rispettivamente:

- LoginLDAP;
- RetrievalInfoLDAP;
- UtenteModificationsLDAP.

[vi] Jakarta Hivemind è un broker di servizi per applicazioni Java. Il prodotto è Open Source e mantenuto dalla fondazione Apache. Sorgenti e documentazione disponibili presso http://jakarta.apache.org/hivemind/

5.4. Realizzazione delle pagine

Le classi descritte in precedenza implementano sia la logica applicativa dell'Interfaccia sia la configurazione della logica strutturale delle pagine. La presentazione è tuttavia affidata soltanto a pagine HTML, mappate in Tapestry come componenti configurati per interagire con il codice sottostante.

5.4.1. Specifica delle mappature in Tapestry

Il processo di realizzazione di pagine web per il rendering con Tapestry è decisamente semplice e lineare. Il framework consente di realizzare pagine template utilizzando qualsiasi editor HTML per ottenere pagine statiche. In seguito è possibile collegare alcuni tag delle pagine a componenti di Tapestry specificando una proprietà aggiuntiva del tag che viene ignorata dalla maggioranza degli editor e dei browser. In sede di rendering, Tapestry identificherà i tag con la proprietà specificata ed eseguirà le procedure prescritte sostituendoli con il componente su cui sono mappati. Questa procedura differisce totalmente da quelle normalmente adottate per lo sviluppo web con sistemi come PHP o JSP[vii], infatti offre allo sviluppatore la possibilità di visualizzare immediatamente l'output della pagina prima di eseguire il rendering, mediante il template.

Questa strategia di sviluppo impone attenzione sul modo in cui i tag da associare a componenti sono utilizzati da Tapestry. Non importa che tipo di tag sia presente sul template HTML, in molti casi Tapestry lo sostituirà con il tag previsto per il componente. In questo senso, il tag , normalmente utilizzato per porzioni di testo, se associato al componente Input verrebbe sostituito da un campo di testo a inserimento, ovvero il tag HTML <input>. Alcuni componenti in Tapestry conservano il corpo, ovvero il tag nel quale sono dichiarati, altri lo sostituiscono totalmente. Lo sviluppatore deve fare attenzione a quali tag non conservano il corpo per accertarsi di non affidare al corpo informazioni di rilievo per il foglio di stile[viii] della pagina. Ad esempio, una buona politica è includere sempre i tag associati a compo-

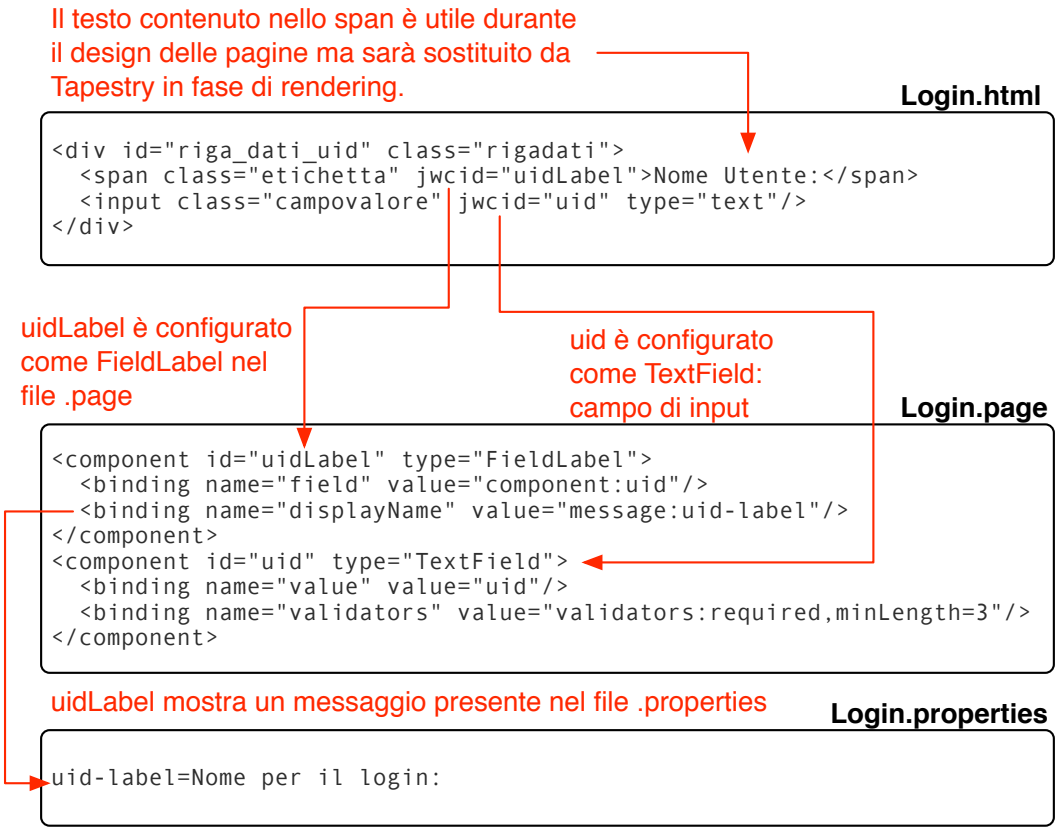
[vii] Rispettivamente PHP: Hypertext Processor e Java Server Pages (JSP). Si tratta di due tecnologie per la realizzazione di siti web dinamici. Maggiori informazioni presso http://www.php.net/ e http://java.sun.com/products/jsp/

[viii] CSS o Cascading Syle Sheet. Si tratta di un meccanismo semplice e standard per aggiungere informazioni stilistiche a pagine web. La tecnologia è mantenuta presso il World Wide Web Consortium: http://www.w3.org/Style/CSS/

nenti di Tapestry in tac strutturali (come il in XHTML o per il testo) e affidare le informazioni per la visualizzazione a questi tag.

La mappatura con componenti di Tapestry viene eseguita specificando la proprietà jwcid=”id_elemento”. Ogni id_elemento è mappato in un file XML con lo stesso nome della pagina ed estensione .page. In questo file xml vengono specificate le caratteristiche di ogni mappatura, come elementi a cui fa riferimento nel codice oppure proprietà statiche, come campi di testo. Accompagnato al file.page è talvolta presente un file .properties con lo stesso nome della pagina. Questo file contiene le proprietà statiche degli elementi, come il testo da visualizzare sulle etichette. Attraverso i file .properities è possibile specificare più di una localizzazione linguistica per ogni pagina, da scegliere a tempo di esecuzione in base alla lingua del browser. La semantica .html .page .properties può essere condensata con specifiche nel file HTML, attraverso proprietà specificate nei tag da renderizzare come componenti di tapestry. Nel corso della realizzazione è stata utilizzata la specifica estesa in ogni occasione.

Figura 5.5: Esempio di specifica delle mappature: pagina Login

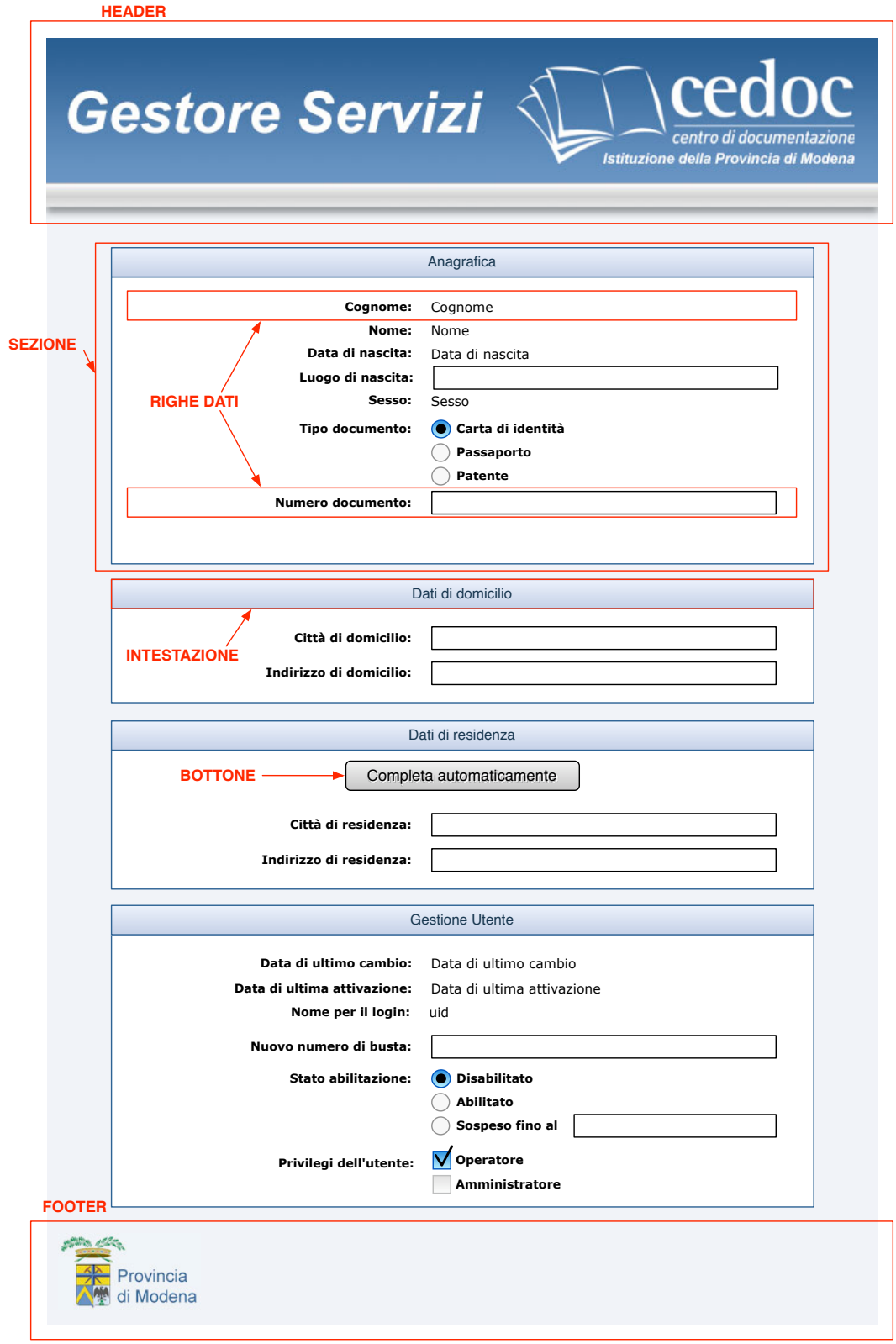


5.4.2. Composizione

In conformità con l'architettura a componenti di Tapestry e con gli standard XHTML e CSS, le pagine sono costituite da un insieme di componenti annidati, suddivisi genericamente in componenti strutturali o non mappati inTapestry e componenti attivi, ovvero mappati in Tapestry. I componenti strutturali portano le informazioni necessarie a identificare parti di ogni pagina per l'applicazione del foglio di stile.

La pagina è genericamente costituita da alcuni contenitori annidati: esiste il contenitore principale con all'interno l'intera pagina, un contenitore superiore utilizzato per la grafica del "header" e una serie di contenitori denominati genericamente sezioni per suddividere le informazioni in modo logico ma anche fisico nella visualizzazione. Ogni sezione è composta da un insieme di righe, ciascuna delle quali può avere contenuti arbitrari all'interno. La riga più comune dispone di un'etichetta e di un campo di input. Questa è la struttura generale utilizzata per il modulo di inserimento dati. Ogni pagina è conclusa da un contenitore inferiore per la grafica della parte bassa della pagina (footer).

Figura 5.6: Estratto della pagina Utente



Nell'esempio è possibile identificare chiaramente le sezioni. Ogni sezione ha sfondo bianco ed è intestata da una riga in sfondo blu, contenente il titolo della sezione. All'interno della sezione sono presenti più righe, completate dai campi utili all'interazione con la pagina. Nella logica della presentazione il Tipo di Documento con le tre opzioni è ospitato da una sola riga. Ereditare la struttura in ogni pagina consente di realizzare un unico foglio di stile e rende molto più veloce la mappatura dei componenti attivi.

5.4.3. Componenti per etichette e campi di input

La maggior parte dei componenti utilizzati per il progetto implementa i moduli di inserimento dati. Tapestry offre due componenti predefiniti per realizzare un modulo di inserimento composto da numerosi campi di input e relative etichette.

Il campo di input è realizzato attraverso il componente di Tapestry "TextField". Tipicamente questo componente viene configurato associando il percorso OGNL di una variabile all'interno del codice della pagina. Quando un TextField fa parte di un Form, il form submit assocerà il valore contenuto nel campo alla variabile. E' possibile specificare alcuni validator, i quali si occuperanno della verifica del testo inserto.

A un TextFiled è possibile associare una FeldLabel. Una FieldLabel è un semplice campo di testo con una estensione particolarmente utile. L'associazione dell'etichetta con un TextField consente di utilizzare l'etichetta stessa per segnalare un errore di validazione del contenuto. In altre parole, se l'eventuale validator specificato nella configurazione del TextField dà esito negativo è possibile modificare i parametri della FieldLabel associata per segnalare l'errore, ad esempio colorando il testo di rosso.

I restanti campi di testo statici vengono resi attraverso i componenti Insert, utilizzati normalmente per renderizzare testo generico. Gli elementi di pagine di tipo Insert dovrebbero sempre essere dichiarati all'interno di un tag strutturale o di uno . I componenti Insert scartano il corpo del proprio tag, pertanto l'inclusione è necessaria per compatibilità con gli standard di HTML e XHTML.

Segue un estratto della configurazione della pagina Utente che fa uso dei tre componenti

Da Utente.html

```
<!-- Intestazione della sezione gestione utente e password -->
<div class="intestazione"><span jwcid="passHead">Gestione Utente</span></div>

<!-- ... -->

<!-- Etichetta e campo di input per la tessera da associare all'utente -->
<div id="riga_dati_busta" class="rigadati">
  <span id="etichetta_busta" class="etichetta" jwcid="envSLabel">Numero Busta:</span>
  <input class="campovalore" id="busta" jwcid="envS" type="text" value="#" onfocus="cardNumI-
sValid = false"/>
</div>
```

Da Utente.page

```
<!-- Intestazione della sezione gestione dati utente e password, esempio di intestazione con
uso di Insert per associare un testo preso da Utente.properties -->
<component id="passHead" type="Insert">
  <binding name="value" value="message:pass-head"/>
  <binding name="raw" value="true"/>
</component>

<!-- ... -->

<!-- Etichetta e campo di input per l'inserimento della busta da associare all'utente. La va-
lidazione è lato server attraverso Envelope. In caso di errore l'etichetta diventa rossa -->
<component id="envSLabel" type="FieldLabel">
  <binding name="field" value="component:envS"/>
  <binding name="displayName" value="message:envS-label"/>
  <binding name="raw" value="true"/>
</component>
<component id="envS" type="TextField">
  <binding name="value" value="ognl:associatedEnvelope"/>
  <binding name="validators" value="validators:$envelopeValidator"/>
</component>
```

Da Utente.properties

```
#Localizzazioni in italiano dei campi configurati in precedenza
pass-head=Gestione utente
envS-label=Numero busta:
```

5.4.4. Componenti per oggetti ciclici e opzionali

Pagine come Risultato e Duplicati devono poter visualizzare un numero di elementi variabile. La logica di Tapestry per la realizzazione del template HTML viene incontro all'esigenza di realizzare pagine di questo tipo con due componenti predefiniti.

Il template deve essere definito con un campione dell'oggetto da replicare, racchiuso all'interno di un tag dichiarato come componente For in Tapestry. L'intero contenuto del tag deve fare riferimento al foglio di stile per classe e non per id, allo scopo di favorire la moltiplicazione conservando lo stile. Nel codice della pagine è possibile associare la visualizzazione di un numero arbitrario di elementi, ciascuno dei quali può contenere componenti dinamiche il cui valore viene compilato diversamente per ciascun elemento. La pagina Ricerca opera per esempio con una lista di entità utente e genera un oggetto di riepilogo per ciascun elemento della lista.

I componenti opzionali possono essere racchiusi in tag mappati come componenti If e Else. In questo modo è possibile specificare attraverso il codice se il componente deve o meno essere renderizzato al momento del richiamo della pagina. Per realizzare la notifica degli errori, delle informazioni e dei successi, si è scelto di includere tre campi nella parte superiore della pagina, ciascuno racchiuso in un tag If. Attraverso il codice è possibile specificare quale campo visualizzare a seconda degli eventi che occorrono nell'esecuzione. Queste scelte influiscono sul caricamento della pagina: in Tapestry 4 non è possibile utilizzare componenti ciclici o opzionali associati a interventi lato client con JavaScript.

Un esempio di uso di un componente opzionale è nella pagina Utente. In questa pagina deve essere possibile inserire un valore per il garante o genitore soltanto se l'utente è minorenne. La riga dati per l'inserimento dei valori, contenente etichetta e campo di input, è racchiusa in un componente che ne determina la visualizzazione.

Da Utente.html

```
<!-- Tutto ciò che vive all'interno del <div> con jwcid showGarante è a visualizzazione opzionale -->
<div jwcid="showGarante">
  <div id="riga_dati_patria_potesta" class="rigadati">
```

```
    <span id="etichetta_patria_potesta" class="etichetta" jwcid="garanteLabel">Patria ptest&a-
grave;:
    </span>
    <input class="campovalore" id="patria_potesta" jwcid="garante" type="text" value="#" />
  </div>
</div>
```

Da Utente.page

```
<!-- showGarante è mappato in un componente If, per gestire la visualizzazione opzionale del
<div> e di tutto il suo contenuto -->
<component id="showGarante" type="If">
  <binding name="condition" value="showGarante"/>
</component>
<component id="garanteLabel" type="FieldLabel">
  <binding name="field" value="component:garante"/>
  <binding name="displayName" value="message:garante-label"/>
  <binding name="raw" value="true"/>
</component>
<component id="garante" type="TextField">
  <binding name="value" value="ognl:person.garante"/>
  <binding name="validators" value="validators:$garanteValidator"/>
</component>
```

Un esempio di utilizzo del componente For è presente nella pagina per la visualizzazione dei risultati della ricerca, Risultato:

Da Risultato.html

```
<!-- resEnv deve comparire una volta per ciascun risultato. I valori dei jwcid cn, birthDate,
tessera e edit devono essere impostati per ogni riproduzione di resEnv -->
<div jwcid="resEnv">
  <div class="risultato">
    <div class="infolabel">
      <span jwcid="cn">Canonical Name</span>
    </div>
    <div class="infolabel">
      <span jwcid="birthDateLabel">Data di nascita:</span>
      <span jwcid="birthDate">Data nascita</span>
    </div>
    <div class="infoLabel">
      <span jwcid="tesseraLabel">Ultima tessera bibliotecaria nota:</span>
      <span jwcid="tessera">12345678</span>
    </div>
    <div class="editbutton">
      <input jwcid="edit" type="button" value="Modifica Dati"/>
    </div>
  </div>
```

```
</div>
```

Da Risultato.page

```
<!-- resEnv è dichiarato come for, gli altri elementi utilizzano la stessa logica vista in
precedenza -->
<component id="resEnv" type="For">
  <binding name="source" value="ognl:persons"/>
  <binding name="value" value="ognl:person"/>
</component>
<component id="cn" type="Insert">
  <binding name="value" value="ognl:person.commonName"/>
</component>
<component id="birthDateLabel" type="Insert">
  <binding name="value" value="message:bdate-label"/>
</component>
<component id="birthDate" type="Insert">
  <binding name="value" value="formattedBirthDate"/>
</component>
<component id="tesseraLabel" type="Insert">
  <binding name="value" value="message:tessera-label"/>
</component>
<component id="tessera" type="Insert">
  <binding name="value" value="ognl:person.lastTessera"/>
</component>
```

5.4.5. Radiobutton e checkbox

Molti form raccolgono informazioni da scegliere tra valori fissi attraverso gli oggetti radiobutton e checkbox. In Tapestry è possibile realizzare entrambi gli oggetti attraverso componenti predefiniti. In particolare, la semantica dei radiobutton vede componenti Radio inclusi in gruppi, componenti RadioGroup. I Radio possono essere associati a una variabile che ne definisce la scelta attraverso percorsi OGNL. I checkbox sono invece indipendenti da gruppi e dichiarati attraverso componenti di tipo Checkbox.

La pagina utente raccoglie esempi di entrambi i costrutti, i componenti Radio sono utilizzati per la selezione del tipo di documento.

Da Utente.html

```
<!-- I tag <span> sono semplici etichette ai radiobutton, i tag <input> sono tutti radiobutton
-->
<div jwcid="rdGroupDoc">
  <span class="etichettabottone" jwcid="idCardLabel">Carta di ID</span>
```

```
  <input jwcid="idCardRd" class="radiobutton" type="radio" value="carta"/>
  <span class="etichettabottone" jwcid="dLicLabel">Patente</span>
  <input jwcid="dLicRd" class="radiobutton" type="radio" value="patente"/>
  <span class="etichettabottone" jwcid="passLabel">Passaporto</span>
  <input jwcid="passRd" class="radiobutton" type="radio" value="passaporto"/>
</div>
```

Da Utente.page

```
<!-- rdGroupDoc è dichiarato RadioGroup. La dichiarazione associa tra di loro i Radio, fornendo funzioni quali la scelta di un valore predefinito impostato dal codice al momento di rendering -->
<component id="rdGroupDoc" type="RadioGroup">
  <binding name="selected" value="ognl:person.docType"/>
  <binding name="validators" value="validators:tipoDoc"/>
</component>
<component id="docTypeLabel" type="Insert">
  <binding name="value" value="message:docType-label"/>
  <binding name="raw" value="true"/>
</component>
<component id="idCardLabel" type="Insert">
  <binding name="value" value="message:idCard-label"/>
  <binding name="raw" value="true"/>
</component>
<component id="dLicLabel" type="Insert">
  <binding name="value" value="message:dLic-label"/>
  <binding name="raw" value="true"/>
</component>
<component id="passLabel" type="Insert">
  <binding name="value" value="message:pass-label"/>
  <binding name="raw" value="true"/>
</component>

<!-- I tre Radio hanno valore legato al path OGNL di tre attributi descritti dall'interfaccia ICeDocPersonalData -->
<component id="idCardRd" type="Radio">
  <binding name="value" value="ognl:@dps.model.ICeDocPersonalData@CARTA_IDENTITA"/>
</component>
<component id="dLicRd" type="Radio">
  <binding name="value" value="ognl:@dps.model.ICeDocPersonalData@PATENTE"/>
</component>
<component id="passRd" type="Radio">
  <binding name="value" value="ognl:@dps.model.ICeDocPersonalData@PASSAPORTO"/>
</component>
```

Il checkbox è utilizzato per l'abilitazione degli utenti come operatori.

Da Utente.html

```
<div class="rigabottoni">
  <span class="etichettabottone" id="etichetta_carta" jwcid="opLabel">Abilita l'utente come
OPERATORE</span>
  <input type="checkbox" class="check" jwcid="opCb"/>
</div>
```

Da Utente.page

```
<component id="opLabel" type="FieldLabel">
  <binding name="displayName" value="message:op-label"/>
  <binding name="raw" value="true"/>
  <binding name="field" value="component:opCb"/>
</component>

<!-- Il valore del checkbox è associato a operatorStatus -->
<component id="opCb" type="Checkbox">
  <binding name="value" value="operatorStatus"/>
</component>
```

5.4.6. Componenti per la gestione del form

Tapestry conserva la semantica del form HTML come un contenitore di campi, pulsanti e opzioni che può essere inviato al server utilizzando il metodo POST del protocollo HTTP. Per implementare la semantica del form nella pagina renderizzata, Tapestry utilizza il componente Form. Il componente Form cattura il valore di ogni RadioGroup, CheckBox o TextField al suo interno al momento del submit. Nella pagina finale Tapestry scrive il JavaScript necessario a implementare qualsiasi tipo di validazione lato client imposta dalla configurazione dei componenti. In questo senso, gli eventi JavaScript associati all'azione di submit sono completamente personalizzabili attraverso il framework, ad esempio modificando il codice del validator che si occupa della gestione di uno o più componenti.

Nella stesura della pagina è sufficiente realizzare un tag strutturale da mappare sul componente Form, Tapestry manterrà il corpo del tag e assocerà l'apparato di invio a questo tag. Ogni pagina che richiede l'invio di dati utilizza questa dichiarazione.

Il submit del form è regolato da componenti dichiarati Submit (per bottoni semplici) o ImageSubmit (per bottoni da sostituire con immagini). Al momento del rendering della pagina, tapestry sostituirà i componenti Submit con tag <input> nel file HTML di destinazione.

Un esempio di semplice utilizzo degli elementi Form è nella pagina Login: l'utente deve fornire un nome e una password per proseguire. La pagina dispone di due bottoni, uno esegue il submit, l'altro è dichiarato PageLink e serve per riferirsi direttamente a un'altra pagina. Nella fattispecie il PageLink è etichettato "Annulla" e viene renderizzato come un tag <a href> che punta alla pagina precedente.

Da Login.html

```
<!-- L'intero blocco di inserimento è racchiuso in un form. Il submit è regolato dal bottone
con id "submit" -->
<form id="login" jwcid="login" method="post">
  <div id="nome_pwd" class="sezione">
    <div class="intestazione"><span jwcid="uidHead">Nome e Password</span></div>
    <div id="riga_dati_uid" class="rigadati">
      <span class="etichetta" jwcid="uidLabel">Nome Utente:</span>
      <input class="campovalore" jwcid="uid" type="text"/>
    </div>
    <div id="riga_dati_uid" class="rigadati">
      <span class="etichetta" jwcid="userPasswordLabel">Password:</span>
      <input class="campovalore" jwcid="userPassword"/>
    </div>
  </div>
  <div id="bottoni" class="sezione">
    <div id="riga_submit" class="rigatasti">
      <input id="submit" jwcid="submit" value="Procedi"/>
      <!-- Il tag è dichiarato <a> solo per facilità di visualizzazione in sede di disegno.
Tapestry sostituisce <a></a> la momento del rendering. L'immagine è invece fondamentale perché
viene mantenuta all'interno del componente. -->
      <a jwcid="toStart" href="http://servizi.cedoc.mo.it">
        <img src="img/but_small_cancel.jpg"/>
      </a>
    </div>
  </div>
</form>
```

Da Login.page

```
<!-- Un asset crea un oggetto personalizzato referenziabile direttamente dai file .page. In
questo caso si fa riferimento a un'immagine da utilizzare come segnaposto per l'ImageSubmit
più in basso -->
<asset name="proceed" path="/img/but_small_proceed.jpg"/>
```

```
<!-- Il Form è associato a un listener implementato in un metodo della classe e abilita la
validazione -->
<component id="login" type="Form">
  <binding name="success" value="listener:attemptLogin"/>
  <binding name="stateful" value="false"/>
  <binding name="clientValidationEnabled" value="false"/>
</component>
<component id="uidHead" type="Insert">
  <binding name="value" value="message:uid-head"/>
</component>
<component id="uidLabel" type="FieldLabel">
  <binding name="field" value="component:uid"/>
  <binding name="displayName" value="message:uid-label"/>
</component>
<component id="uid" type="TextField">
  <binding name="value" value="uid"/>
  <binding name="validators" value="validators:required,minLength=3"/>
</component>
<component id="userPasswordLabel" type="FieldLabel">
  <binding name="field" value="component:userPassword"/>
  <binding name="displayName" value="message:password-label"/>
</component>
<component id="userPassword" type="TextField">
  <binding name="value" value="userPassword"/>
  <binding name="validators" value="validators:required,minLength=3"/>
  <binding name="hidden" value="true"/>
</component>

<!-- L'ImageSubmit viene sempre associato alla funzione submit del form. La configurazione
riguarda soltanto l'immagine da visualizzare per segnaposto -->
<component id="submit" type="ImageSubmit">
  <binding name="image" value="asset:proceed"/>
</component>

<!-- Il PageLink viene renderizzato come <a>. Il vantaggio rispetto all'uso diretto di <a> è
che il PageLink viene compilato a tempo di esecuzione, quindi è tollerante alle modifiche sul-
le posizioni delle pagine -->
<component id="toStart" type="PageLink">
  <binding name="page" value="literal:Start"/>
</component>
```

5.5. Formazione del personale

Fin dalle ricerche preliminari per la realizzazione del sistema è apparso evidente che la formazione del personale sarebbe stata essenziale per la corretta messa in funzione delle pro-

cedure. La realtà gestita dal CeDoc è infatti caratterizzata da personale addetto a servizi bibliotecari con una limitata conoscenza informatica. Tutto il personale della biblioteca ricade in questa categoria e costituisce il gruppo di utenti considerati Operatori. Ciò significa che non solo gli operatori devono essere preparati alle eventualità di gestione ma devono essere anche in grado di spiegare a gli utenti le procedure necessarie per il nuovo sistema di collegamento a internet.

La percezione del sistema da parte degli operatori è limitata alle procedure o al modo in cui le nuove infrastrutture si interfacciano con gli strumenti software con cui hanno familiarità e alle procedure legate all'interfaccia web. Non è necessario comunicare dettagli implementativi a tutti gli operatori, anche se si è rivelato molto utile consultare alcuni di essi durante varie fasi del progetto.

Il vero e proprio sforzo di formazione si divide in due fasi:

- Realizzazione di un manuale completo;
- Promozione e svolgimento di corsi interni per l'utilizzo del programma.

5.5.1. realizzazione di un manuale operativo

Nel momento in cui le caratteristiche principali dell'interfaccia web hanno raggiunto un livello definitivo, è stato possibile redarre un manuale. Lo scopo del manuale era informare gli operati di tutte le procedure che avrebbero dovuto applicare una volta reso operativo il sistema.

Per non rendere l'applicazione troppo complessa le procedure sono state semplificate e formalizzate in alcuni passaggi definiti, descritti con dovizia di particolari e ridondanza dei concetti chiave. Il manuale è stato realizzato per fornire una guida nelle prime fasi di utilizzo del prodotto, in seguito gli operatori sono stati abbastanza indipendenti da non doverlo più consultare. La struttura a procedure del manuale lo rende adatto alla consultazione contemporanea alle attività del sistema come guida passo passo alla procedura.

La stesura del manuale è stata progressivamente raffinata nel corso del perfezionamento del sistema e dello svolgimento dei corsi. In particolare, si è scelto di corredare al manuale una sezione di domande frequenti, completa di tutte le domande più rilevanti raccolte dall'esperienza dei corsi, unite ad alcune domande tipiche degli operatori. La sezione domande frequenti si è rilevata particolarmente utile per fissare alcuni concetti fondamentali delle procedure che potevano essere stati trascurati e considerati di minore importanza. Gli operatori hanno fatto uso della sezione per verificare alcuni dubbi senza dover fare la domanda direttamente all'ufficio tecnico.

Malgrado la sua utilità, il manuale da solo non avrebbe mai potuto garantire un livello di preparazione accettabile; esso resta tuttavia la base per l'inserimento di nuovi operatori che vengono abilitati dai colleghi e devono recepire rapidamente i concetti fondamentali senza il più corposo aiuto dei corsi di formazione.

5.5.2. Corsi di formazione

Sebbene decisamente complessi da realizzare e da gestire, i corsi di formazioni si sono rivelati fondamentali per alcuni motivi:

- Hanno consentito agli operatori di toccare con mano in anticipo l'intera procedura di base e di cominciare a prendere confidenza con gli strumenti nuovi a disposizione;
- Hanno favorito lo sviluppo di interfacce adatte alle esigenze degli operatori, espresse man mano che i corsi procedevano;
- Sono stati un'eccellente banco di prova per rilevare errori o imprecisioni nel sistema.

In totale si sono tenute sei lezioni tutte uguali. Ogni lezione in un tempo massimo di quattro ore doveva trasmettere i concetti fondamentali e possibilmente dare la possibilità a ogni operatore di provare il nuovo sistema. La prima lezione è stata anticipata molto rispetto alle altre e ha presentato una versione beta dell'intero software. Lo scopo dell'anticipo era scegliere un momento adeguato in cui rilevare un feedback fondamentale da parte degli operatori. Era necessario raggiungere il personale con un prodotto funzionante eppure tale da consentire modifiche non eccessivamente complesse.

In seguito sono state effettuate quattro lezioni uguali, con ridotto o nullo sviluppo del software tra una lezione e l'altra. Il software aveva raggiunto un discreto livello di usabilità per queste lezioni, il cui scopo era quello di aggragare più personale possibile per la formazione su procedure quasi definitive.

L'ultima lezione non era prevista dal programma originario e si è svolta ritardata rispetto al blocco centrale. Lo scopo di questa lezione era quello di finalizzare la raccolta di domande e di completare la formazione di tutti gli operatori autorizzati.

Nel corso di ogni lezione sono stati raccolti i nomi dei partecipanti per la prima abilitazione. La prima abilitazione degli operatori si è svolta presso il CeDoc ed è stata limitata ai soli operatori che avevano sostenuto il corso. I successivi sono stati abilitati direttamente dai colleghi. I primi utenti sono stati abilitati dagli operatori in biblioteca.

Ogni corso è stato strutturato in alcune fasi:

- Introduzione al nuovo sistema di autenticazione. Concetto di autenticazione, motivi del cambiamento, dimostrazione del sistema di autenticazione;
- Nuovi strumenti: descrizione e spiegazione del funzionamento dei nuovi strumenti. Dimostrazione di utilizzo degli strumenti;
- Un esempio di procedura: iscrizione e abilitazione di un nuovo utente partendo da Auriga, dimostrazione e prova pratica per tutti - Pausa;
- Gestione degli errori: descrizione degli errori possibili e simulazione di eventi insoliti;
- Procedure aggiuntive e scadenze temporali: descrizione e dimostrazione delle procedure aggiuntive, prova pratica per tutti.
- Riepilogo: riepilogo procedure e configurazione dei computer.

Nel corso delle lezioni gli operatori hanno recepito le nuove procedure abbastanza in fretta. Alcune eccezioni sono costituite da personale che utilizzava convenzioni procedurali assodate ma non supportate dagli strumenti in uso. Un esempio è l'utilizzo del campo dedicato al Nome in Auriga per la segnalazione di abilitazioni particolari, come il prestito di audiovisivi. La segnalazione era effettuata apponendo un asterisco (*) al nome dell'utente per co-

modità di visualizzazione nei risultati della ricerca in Auriga. Questa procedura non è mai stata supportata dal client Java del sistema Auriga, pertanto non è stata salvaguardata dalla modifica dovuta alle procedure di iscrizione. Gli operatori che fanno uso di procedure non supportate hanno dovuto adeguarsi al nuovo sistema o richiedere modifiche personalizzate.

6. ESTENSIONI AL PROGETTO BELLEROFONTE

6.1. Interfaccia amministrativa

Attualmente le procedure di amministrazione del sistema Bellerofonte coinvolgono l'utilizzo di un applicativo client di terze parti studiato per l'interazione con LDAP. Questo applicativo è un software generico che consente la gestione di sistemi LDAP di elevata complessità ma non può ragionevolmente automatizzare le procedure di manutenzione più comuni, tra cui la verifica amministrativa degli utenti e la risoluzione problemi non identificati.

Per automatizzare le procedure caratteristiche del sistema e rivolte all'infrastruttura LDAP è stata progettata una nuova interfaccia web. L'interfaccia si integrerà in quella già attiva per utenti e operatori ma fornirà le funzionalità avanzate soltanto agli amministratori del sistema. Si riportano alcune funzioni della nuova interfaccia.

6.1.1. Funzioni di ricerca

Le funzionalità di ricerca richieste all'applicativo sono dei seguenti tipi:

- Ricerca di utenti, attivi e duplicati;
- Ricerca di schede con password.

La prima modalità consente di cercare sul ramo ou=utenti-duplicati,o=CeDoc, oltre che su utenti. Per l'input della query di ricerca si propone un sistema a AND che incrocia i risultati dei seguenti filtri:

- Nome: filtro di ricerca sull'attributo givenName;
- Cognome: filtro di ricerca sull'attributo sn;
- Tessera bibliotecaria: numero completo della tessera bibliotecaria, comprensivo di eventuali prefissi;

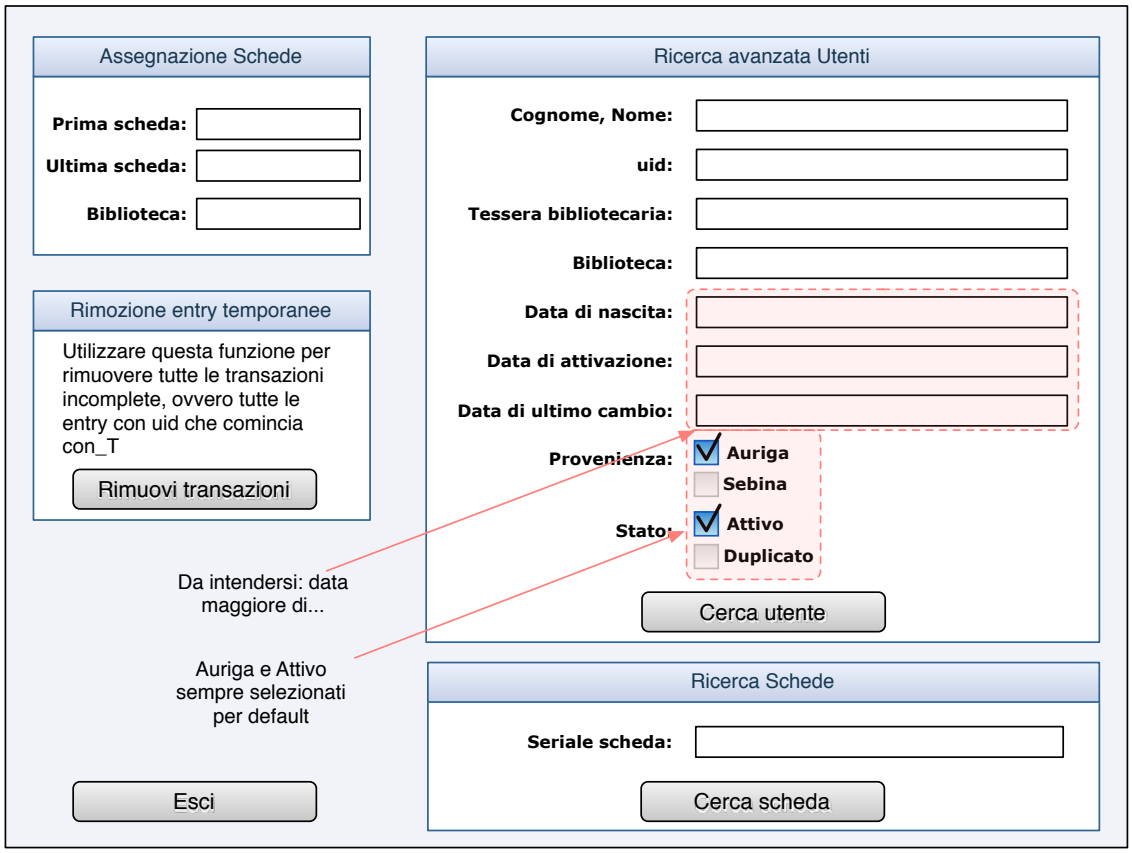
- Data di nascita: filtro di ricerca sull'attributo CeDocMoDataNascita. La ricerca su questo attributo deve essere esplicitata nella forma di "maggiore o uguale a". Questo consentirà di recuperare immediatamente tutte le entry corrispondenti a utenti minorenni o maggiorenni di entrambi i rami;
- Stato di attivazione: attributo aggiuntivo, se lo stato di attivazione nel modulo di ricerca viene impostato su "non attivo" il sistema ricercherà negli uid che cominciano con _T, ovvero gli uid "di transazione", corrispondenti alle sole transazioni pendenti;
- Data ultimo cambio e ultima attivazione: filtri per gli attributi CeDocMoDataUltimoCambio e CeDocMoDataUltimaAttivazione. Utili per determinare tutti gli utenti che devono cambiare la loro password a breve o richiedono attenzione. Anche questi filtri devono consentire la ricerca nella forma di "maggiore o uguale a";
- Servizio di Provenienza: filtro sull'attributo CeDocMoUltimoServizio, utile per cercare solamente utenti appartenenti ad Auriga o Sebina;
- Biblioteca di appartenenza (la funzionalità richiede di intervenire in lettura sul ramo ou=biblioteche,o=CeDoc).

La query di ricerca è pertanto composta dall'intersezione di tutte le condizioni per cui l'utente specifica un valore di confronto.

La prima modalità di ricerca opera nello stesso modo previsto per la ricerca di utenti nell'interfaccia per operatori.

La seconda modalità di ricerca ha la funzione di raggiungere una o più schede in base al seriale della scheda. L'obiettivo di questa funzionalità e consentire il reperimento della scheda per verificare la data di assegnazione, l'operatore che ha confermato l'assegnazione e l'assegnatario della scheda. La ricerca opererà con il solo campo del seriale come termine di confronto. La visualizzazione delle schede associate a un utente è comunque possibile attraverso la ricerca degli utenti.

Figura 6.1: Interfaccia amministrativa, maquette della pagina principale



6.1.2. Funzioni batch

Il sistema dovrà consentire l'avvio manuale di job batch gestiti dal bus java. L'interfaccia deve prevedere i seguenti job:

- Rimozione delle entry non attivate dal ramo ou=utenti,o=CeDoc: le entry non attivate corrispondono a transazioni catturate dal bus ma non ancora validate dai bibliotecari. Queste transazioni corrispondono a utenti nello stato "Temporaneo" il cui uid comincia con la keyword "_T". La composizione dell'uid consente di determinare l'età relativa alla transazione. Il job di rimozione coinvolge soltanto le transazioni più vecchie di 4 ore. Il job deve anche occuparsi di rimuovere qualunque entry associata all'utente rimosso dal ramo ou=tessere,o=CeDoc;
- Assegnazione automatica delle schede alle biblioteche: questa funzione consente all'amministratore di completare l'assegnazione di un range di seriali che identificano schede

con password a una biblioteca. L'amministratore deve specificare il primo e l'ultimo seriale del range, la sigla della biblioteca assegnataria e confermare l'operazione.

6.1.3. Funzioni di modifica e validazione

Per favorire l'analisi dei dati ottenuti dalla ricerca, la pagina dei risultati non deve rimandare in nessun caso alla pagina dei duplicati, pertanto non avviene alcuna verifica sulle condizioni di duplicazione formulate per il comportamento dell'interfaccia per operatori. La richiesta di modifica dei dati di un utente deve ricondurre alla versione per amministratori della pagina di riepilogo dei dati dell'utente, la quale deve consentire le seguenti operazioni:

- Visualizzazione e modifica dell'intera anagrafica dell'utente. La sezione anagrafica non riporterà campi nascosti come quella accessibile dagli operatori e tutti i campi saranno modificabili. La validazione dei dati inseriti nella sezione anagrafica sarà effettuata soltanto sul valore "Patria potestà", per evitare danni involontari al sistema;
- Visualizzazione dell'attributo uid dell'utente, senza possibilità di modifica;
- Visualizzazione dell'uid suggerito per l'utente: il sistema deve prevedere una funzione di generazione dell'uid per l'utente che possa essere azionata manualmente nel caso l'utente visualizzato corrisponda a una entry temporanea. Se la funzionalità non viene attivata e l'uid non viene generato non sarà possibile eseguire altre operazioni che agiscano in scrittura sul database;
- Visualizzazione e modifica di tutti i dati di residenza e domicilio, validata solo per l'esistenza dei dati di residenza;
- Reset della password dell'utente: la password non viene mai visualizzata ma è possibile eseguire un reset inserendo un seriale valido nel campo dedicato all'assegnazione di una nuova scheda;
- Visualizzazione e modifica della data di ultimo cambio e sola visualizzazione della data di ultima attivazione, con validazione prima dell'immissione nel sistema;
- Visualizzazione e modifica dello stato di abilitazione ai servizi, con validazione nella sola data di fine sospensione (controllo che sia una data futura). Allo stato di abilitazione deve essere aggiunta la possibilità di abilitare l'utente come amministratore, in modo distinto

dall'abilitazione come operatore. L'abilitazione come amministratore consente all'utente di operare nell'interfaccia amministrativa;

- Visualizzazione dell'ultimo operatore che ha confermato la modifica dei dati dell'utente;
- Visualizzazione di tutte le tessere bibliotecarie associate all'utente, per ciascuna tessera si visualizzeranno tutti i valori noti per gli attributi: CeDocMoNumeroTessera, CeDocMoNomeServizioTessera, CeDocMoProprietarioTessera (quest'ultimo può essere multiplo). Per ottenere questi dati è necessario eseguire una query sul ramo ou=tessere,o=CeDoc utilizzando come chiave il valore di CeDocMoUltimaTessera del profilo utente. Se i valori non sono disponibili sarà presentato un campo vuoto. Se esiste più di una tessera bibliotecaria associata all'utente visualizzato, deve essere possibile rimuovere l'associazione per tutte le tessera tranne quella il cui valore di CeDocMoNumeroTessera coincide con il valore di CeDocMoUltimaTessera nel profilo dell'utente. Per rimuovere l'associazione si intende cancellare la entry dal ramo ou=tessere,o=CeDoc;
- Visualizzazione di tutte le schede (buste) assegnate all'utente. Questa funzionalità richiede una ricerca aggiuntiva sul ramo ou=schede,o=CeDoc utilizzando come lemma l'uid dell'utente. Di ciascuna scheda saranno visualizzati tutti i valori degli attributi CeDocMoSerialeScheda, CeDocMoDataConsegnaUtente, CeDocMoIdDestinatario, CeDocMoIdOperatoreScheda, se disponibili. Se i valori non sono disponibili sarà presentato un campo vuoto.
- Visualizzazione di tutti gli utenti per i quali sussiste una situazione di duplicazione su una delle condizioni già specificate nel diagramma per la gestione delle transazioni. Nel caso esistano utenti sotto queste condizioni, deve essere consentita la risoluzione automatica del duplicato, intendendo come scelta implicita il mantenimento dell'utente visualizzato e l'applicazione delle politiche per la gestione dei duplicati a tutti gli altri. L'applicazione di tali politiche non applica in nessun caso la verifica sull'attributo CeDocMoProtetto, descritta nel paragrafo successivo e non si applica se l'utente selezionato si trova nello stato "Temporaneo".

La pagina di riepilogo deve inoltre segnalare la posizione dell'utente selezionato. Il DN ottenuto dalla query può appartenere al ramo ou=utenti,o=CeDoc oppure al ramo ou=utenti-

duplicati,o=CeDoc. Solo nel secondo caso può esistere la necessità di riportare il DN nel ramo ou=utenti,o=CeDoc, per intervenire su un errore nella gestione dei duplicati.

Deve essere sempre possibile comandare lo spostamento incondizionato dell'utente nel ramo utenti-duplicati.

6.1.4. Funzioni di spostamento e ripristino

Per funzione di spostamento si intende la possibilità di posizionare utenti nel ramo ou=utenti-duplicati,o=CeDoc. Questa funzione viene richiamata come conseguenza di un comando diretto e incondizionato da parte di un Amministratore. La entry corrispondente all'utente viene spostata e ogni entry del ramo ou=tessere,o=CeDoc che fa riferimento all'utente viene rimossa. Se le entry di tipo tessera candidate per la rimozione fanno riferimento a più di un utente oltre a quello in corso di spostamento, la rimozione si limita alla cancellazione del riferimento al suddetto utente, le entry non vengono cancellate.

Per ripristino si intende l'inverso dello spostamento, un DN viene spostato dal ramo ou=utenti-duplicati,o=CeDoc al ramo ou=utenti,o=CeDoc. Le informazioni disponibili presso il profilo dell'utente consentono sempre di risalire a dati sufficienti per rigenerare una tessera valida, nella fattispecie:

- CeDocMoNumeroTessera può essere derivato dall'attributo CeDocMoUltimaTessera del profilo utente;
- CeDocMoNomeServizioTessera può essere derivato da CeDocMoUltimoServizio nel profilo utente;
- CeDocMoProprietarioTessera è l'uid dell'utente, senza condizioni.

In caso l'utente sia stato spostato per la risoluzione di un duplicato per tessera bibliotecaria, la procedura di ripristino potrebbe generare una tessera già presente nel sistema. In questo caso la procedura di ripristino non è possibile, l'amministratore deve essere avvertito di conseguenza.

Soltanto nel caso di ripristino di utenti gestiti dalla procedura di risoluzione duplicati, deve essere impostato l'attributo CeDocMoProtetto a TRUE. Questo attributo verrà considerato nelle successive operazioni di gestione duplicati per proteggere l'utente da ulteriori spostamenti soltanto se la duplicazione è relativa alla verifica sugli attributi dell'utente (e non alla rilevazione di tessere con molteplici proprietari).

6.2. BiblioMedia

L'efficace infrastruttura di gestione degli utenti realizzata per il progetto Bellerofonte apre le porte alla realizzazione di numerosi servizi accessori alle biblioteche. Con sviluppo in partenza dall'autunno del 2006 è stato realizzato il progetto per un nuovo servizio di content delivery multimediale chiamato BibllioMedia.

Il servizio verrà attivato a partire dall'autunno 2007 nella biblioteca comunale di Carpi e consentirà agli utenti della biblioteca di ascoltare canzoni e visualizzare filmati appartenenti al catalogo multimediale in prestito presso la biblioteca. Con l'aumentare del numero di biblioteche aderenti al servizio sarà possibile condividere musica e video attraverso l'intera infrastruttura di rete del CeDoc. Si riportano alcune funzioni del progetto BiblioMedia.

Ricerca ed esplorazione

Attraverso un portale web gli utenti potranno esplorare i contenuti presenti sul sistema e scegliere cosa ascoltare o visionare. Gli utenti potranno accedere a una funzione di ricerca semplice e immediata, capace di trovare qualsiasi contenuto in poco tempo; oppure potranno personalizzare la ricerca per ottenere risultati mirati alle loro esigenze, attraverso il modulo di ricerca avanzata.

Ascolto e riproduzione

Il portale offre un'esperienza di ascolto e visualizzazione ideata per soddisfare le esigenze di ogni tipo di utente. Gli utenti potranno ascoltare canzoni mentre esplorano i risultati di una ricerca; oppure potranno accedere a schede dettagliate dei contenuti, complete di ogni informazione disponibile.

Il sistema BiblioMedia integrerà l'innovativo modello a scorrimento che consentirà di fondere ascolto ed esplorazione visuale delle canzoni. Gli utenti esploreranno i contenuti musicali disponibili attraverso le copertine degli album come se li potessero scegliere direttamente dallo scaffale.

Album e contenuti

All'interno del portale i contenuti saranno raggruppati in album. Gli utenti potranno cercare contenuti singoli o album. Il passaggio tra un contenuto e l'album di cui fa parte sarà sempre accessibile per favorire l'esplorazione del sistema.

Il sistema consentirà inoltre di ascoltare o visionare l'intero album, attraverso la sua scheda. I contenuti che fanno parte dell'album saranno disponibili nell'ordine di inserimento per la fruizione in sequenza.

Servizi per gli utenti

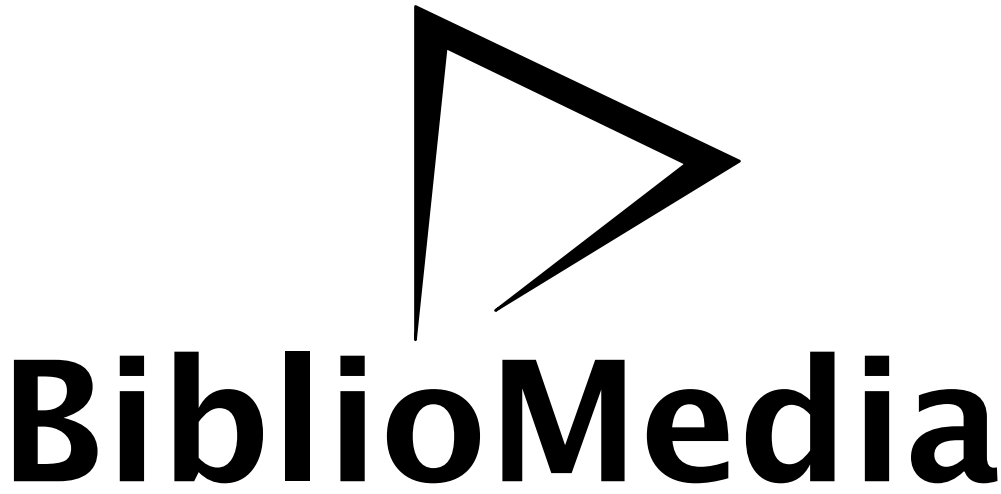
Oltre all'opportunità di accedere a una vasta gamma di contenuti multimediali, Il portale BiblioMedia offrirà agli utenti numerose funzioni.

Gli utenti potranno aggiungere le canzoni o i video che preferiscono a una lista personale alla quale potranno accedere da qualsiasi postazione da cui si possa raggiungere il portale. La lista personale consentirà di elencare rapidamente ogni canzone o video al suo interno, senza doverli cercare di nuovo. Gli utenti potranno inoltre decidere di rendere visibile a tutti gli iscritti la propria lista.

Il sistema progredirà grazie alle preferenze espresse dagli utenti, i quali potranno esprimere un voto per ciascun contenuto, oppure aggiungere commenti e recensioni che renderanno più facile e piacevole l'esplorazione.

Gli utenti potranno anche suggerire agli operatori del sistema quali contenuti acquistare per arricchire l'esperienza di ascolto e riproduzione.

Figura 6.2: Logo del progetto BiblioMedia



Protezione per i minorenni

Il sistema integra una verifica dell'età, che consente di isolare gli utenti minorenni da contenuti non adatti alla loro età, pur senza privarli del piacere dell'esplorazione del resto del catalogo multimediale disponibile. Il controllo sarà invisibile agli utenti grazie alla totale integrazione con il sistema di autenticazione del CeDoc.

Servizi per gli operatori

Il portale BiblioMedia integra un sistema completo di statistiche. Gli operatori del sistema potranno visualizzare e memorizzare dati sulle percentuali di ascolto e visualizzazione, scegliendo di raggrupparli in base a periodi temporali, età utenti e molti altri parametri.

Gli operatori potranno inoltre gestire e mantenere ogni genere di informazione relativa ai contenuti multimediali; potranno gestire i commenti e i suggerimenti pervenuti dagli utenti per mantenere ordinato il sistema.

Servizi di inserimento

Il progetto BiblioMedia comprende la realizzazione di un applicativo autonomo in grado di convertire CD audio e filmati direttamente nel formato utile al sistema multimediale. Il personale addetto all'inserimento dovrà attivare una procedura lineare e automatizzata per preparare contenuti e informazioni correlate all'inserimento.

Risorse

Il portale sarà utilizzabile sia in rete locale, nell'ambito della stessa infrastruttura sia in ambito geografico con la possibilità di distribuire le apparecchiature per lo stoccaggio dei contenuti multimediali su più centri.

L'accesso al sistema sarà regolamentato attraverso il sistema di autenticazione attivo presso il CeDoc, pertanto il database informativo di BiblioMedia non manterrà informazioni personali riguardo agli utenti. Tali informazioni sono conservate separatamente e in accordo con le normative vigenti.

Tecnologie

Il progetto BiblioMedia ha lo scopo di realizzare un prodotto interamente open source e basato su tecnologie leader del mercato. Per la realizzazione è stato scelto un formato audio/video di larga diffusione utilizzabile come componente di interfacce web. Gli utenti non avranno bisogno di altro che un browser web per accedere al portale e alla totalità delle funzioni del sistema.

6.3. Servizi aggiuntivi per gli operatori delle biblioteche

Il progetto Bellerofonte ha raccolto una notevole quantità di informazioni riguardanti non solo gli utenti finali dei servizi in biblioteca ma anche del personale della biblioteca. Le informazioni raccolte hanno portato alla generazione di profili da associare al personale che orbita attorno alla realtà bibliotecaria; profili che possono essere utilizzati per focalizzare servizi generalmente forniti all'entità biblioteca sul singolo individuo. Fanno parte di questi servizi l'accesso alle procedure di prestito e catalogazione e la posta elettronica.

6.3.1. Posta elettronica

L'infrastruttura della posta elettronica del CeDoc prevede un indirizzo e-mail condiviso da tutti gli operatori della biblioteca. In precedenza venivano selezionati alcuni computer ad uso esclusivo degli operatori e configurati per la ricezione della posta. Tutte le macchine ricevevano gli stessi messaggi e operavano con lo stesso account (di conseguenza la stessa password).

L’evoluzione più logica dell’architettura verso una infrastruttura focalizzata sull’utente è la generazione di un indirizzo e-mail “@cedoc.mo.it” per ciascun operatore della biblioteca. La gestione degli indirizzi può essere effettuata direttamente dal server che regola il flusso della posta attraverso l’interazione con il database LDAP.

L’architettura attualmente in elaborazione prevede l’aggiunta di tre attributi a ciascuna entry del ramo ou=utenti,o=CeDoc:

- mail: indirizzo e-mail primario. L’indirizzo viene utilizzato per la consegna dei messaggi e identifica l’operatore per il server di posta;
- mailAlternateAddress: attributo multiplo utilizzato per memorizzare un numero arbitrario di alias. Un alias è un segnaposto per l’indirizzo e-mail e vale come l’indirizzo stesso in sola ricezione;
- mailMessageStore: percorso sul filesystem del server in cui memorizzare i messaggi in ingresso per l’utente.

La procedura di lookup dell’indirizzo per la consegna dei messaggi in arrivo prevede due interrogazioni: una in cerca dell’indirizzo e-mail vero e proprio e una in cerca di alias. In questo modo tutti gli operatori possono avere un indirizzo nominativo del tipo uid@cedoc.mo.it e tutti gli alias necessari a ricevere i messaggi indirizzati alla biblioteca di cui fanno parte, tra cui biblioteche@cedoc.mo.it e nome_biblioteca@cedoc.mo.it.

Le interrogazioni effettuate non insistono sul ramo ou=utenti,o=CeDoc ma su un nuovo ramo denominato ou=mail-accounts,o=CeDoc. Questo ramo ospita prevalentemente entry di tipo alias che puntano a entry del ramo utenti. In OpenLDAP una entry alias è molto simile a un collegamento per i sistemi Windows: si tratta di una entry autonoma della quale fa testo soltanto l’attributo aliasedObjectName che contiene un DN a cui puntare. L’utilizzo di alias consente di realizzare un gerarchia parallela allo spazio nomi.

Il ramo ou=mail-accounts,o=CeDoc ospita anche entry fittizie utilizzate per modellare associazioni del tipo “indirizzo @cedoc.mo.it”-”indirizzo esterno”, ovvero indirizzi e-mail mantenuti per il quali non è presente una mailbox presso il CeDoc.

La gestione degli indirizzi con OpenLDAP facilita notevolmente la migrazione dalla vecchia configurazione caratterizzata su alias scritti su file a una gestione più flessibile e automatizzata.

6.3.2. Accesso nominativo alla catalogazione

I profili degli utenti sono stati realizzati in completa compatibilità con la classe posixAccount. Questa compatibilità rende le entry utilizzabili come utenti UNIX, offrendo numerose opportunità per la configurazione dell'accesso al servizio di catalogazione.

Il servizio attualmente avviene attraverso connessione sicura SSH con una password di accesso generica per la connessione alla macchina e un procedimento di accounting specifico per l'accesso del client di catalogazione.

Attraverso una configurazione basata su LDAP un operatore potrebbe efficacemente accedere al servizio in modo nominativo e autenticato. La m modifica descritta è stata pianificata per modifiche future del sistema Bellerofonte.

7. BACKUP IN SEDE REMOTA

7.1. Processi di Backup

Il tradizionale significato informatico del termine “backup” implica una forma di supporto alle infrastrutture hardware o software di una organizzazione per far fronte a possibili guasti (failures), pur mantenendo la qualità e l’affidabilità del servizio erogato. Possono essere definite procedure di backup sia copie dei dati in locazioni affidabili per definizione, sia ridondanze di servizi o infrastrutture per far fronte a disservizi di quelle considerate principali. Nella fattispecie, è possibile differenziare i metodi di backup, che in ultima analisi coinvolgono i dati informatici, in due tipologie: copie passive e copie attive.

Le copie passive sono copie dell’informazione solamente. Si tratta in genere di copie di sicurezza per evitare la perdita di dati in caso di gravi malfunzionamenti. Le copie passive sono aggiornate periodicamente per mantenere l’attualità dell’informazione ma non sono sufficienti a sostituire repentinamente il servizio che gestiva le informazioni, essendo carenti del servizio stesso. Un esempio di copia passiva è la semplice copia dei dati personali di un utente su un disco rigido esterno o su flash drive. La copia può essere ripristinata se i dati principali risultano danneggiati o persi.

Le copie attive costituiscono il classico esempio di ridondanza. Una copia attiva è in genere una replica di un intero servizio che può essere attivata in caso di malfunzionamento del servizio principale o può essere attiva costantemente per bilanciare il carico tra il servizio principale e quello di backup. Le copie attive sono caratterizzate dalla copia delle informazioni e dalla replica dell’infrastruttura necessaria a gestirle, sebbene possa essere inattiva. Le sincronizzazioni per copia attiva devono essere molto frequenti, per garantire la massima consistenza in caso di guasto. Un esempio di copia attiva è la replica di un database: tutte le informazioni o parte di esse vengono copiate su una versione identica del database che può essere attivata in caso di guasto della prima. I database di grandi dimensioni sfruttano la repli-

cazione a scopo di bilanciamento di carico, distribuendo le interrogazioni attraverso più copie attive.

I due metodi di backup non sono affatto alternativi: spesso si ricorre a replicazione per servizi dei quali è necessario garantire continuità e a backup passivi per informazioni che devono essere memorizzate con riferimento a periodi di tempo molto lunghi. In genere una copia passiva viene utilizzata per accedere a porzioni dell'informazione offline, cioè per analisi di dati memorizzati in momenti successivi al loro utilizzo, anche di mesi o anni.

In questa sezione saranno trattate le caratteristiche del progetto Newbackup, ovvero l'adeguamento del sistema di backup passivo impiegato per l'infrastruttura del CeDoc. Sistemi di replicazione per guasti o bilanciamento di carico sono adottati all'interno dell'infrastruttura ma non sono oggetto di questa trattazione. L'adeguamento è stato reso necessario dal termine della vita utile del precedente sistema e dalle nuove normative in merito al backup in sede remota, ovvero alla replicazione dell'intera infrastruttura di backup in un luogo geograficamente distante per far fronte a gravi danni all'infrastruttura principale.

7.2. Caratteristiche di una strategia di Backup passivo

In merito alla realizzazione di un sistema di stoccaggio dei dati in modo passivo devono essere compiute alcune scelte progettuali per la definizione anzitutto delle policy di backup, ovvero di come l'infrastruttura deve comportarsi. L'implementazione è spesso un fatto secondario, in particolare quando le scelte progettuali sono limitate dalla struttura di sistemi informativi esistenti sulle cui politiche si può intervenire solo marginalmente. Le scelte principali, trattate nel resto di questa sezione, sono:

- Utilizzare un backup incrementale o un backup a snapshot;
- Applicare o meno politiche di replicazione, eventualmente riferite alle sole informazioni o al sistema completo;
- Mantenere una sola copia per ogni dato (la più recente) o realizzare un sistema di copie storiche;

- Consentire al sistema di backup di prelevare i dati o progettare un sistema di stoccaggio passivo;
- Attivare procedure di backup a tempo, pianificate su eventi o su richiesta degli utenti;
- Realizzare un sistema ad hoc per il backup di alcuni servizi o approntare un'infrastruttura di backup per l'intero parco macchine, compresi i computer degli operatori (Simple Backup o Enterprise Grade Backup, NAS e SAN).

7.2.1. Backup incrementale o backup a snapshot

Si suppone che la procedura di backup sia attivata a intervalli regolari. Tra un intervallo e l'altro soltanto una percentuale ridotta della totalità delle informazioni dovrebbe presentare variazioni. Ipotizzando di poter conoscere in tempo utile quali informazioni sono cambiate, è possibile realizzare una forma di backup incrementale, basata sulla memorizzazione di snapshot separati da intervalli temporali elevati e tra uno snapshot e l'altro memorizzare soltanto le variazioni. Gli snapshot sono copie complete della totalità delle informazioni, a ogni intervallo temporale viene memorizzata una nuova versione. Tale struttura prende il nome di backup incrementale e viene spesso utilizzata per il backup rapido di vasti gruppi di computer.

La maggioranza dei prodotti a backup incrementale è largamente distribuita o si presta all'utilizzo di singoli utenti non informatizzati. Risalire alle informazioni appartenenti a un determinato periodo temporale equivale a determinare il più recente snapshot tra quelli precedenti e applicare le modifiche che lo separano dal periodo richiesto. Strategie di backup incrementale rivolte ad ambienti eterogenei sono adottate in numerosi prodotti commerciali e Open source, tra cui BackupPC[i], progetto per il backup di numerosi personal computer. Nel mondo del personal backup, o backup limitato al singolo computer, le strategie di backup incrementale utilizzano sanpshot a versione, ovvero memorizzano ogni modifica quando viene fatta e non a intervalli regolari. Questa strategia si rivela più efficiente per dimensionare il

[i] BackupPC è un progetto mantenuto presso SourceForge per un software di backup centralizzato. Maggiori informazioni presso http://backuppc.sourceforge.net/

backup a singole macchine o addirittura ai dati di singole applicazioni; un buon esempio è lo strumento Time Machine[ii] che sarà integrato dal 2007 nei sistemi operativi Apple.

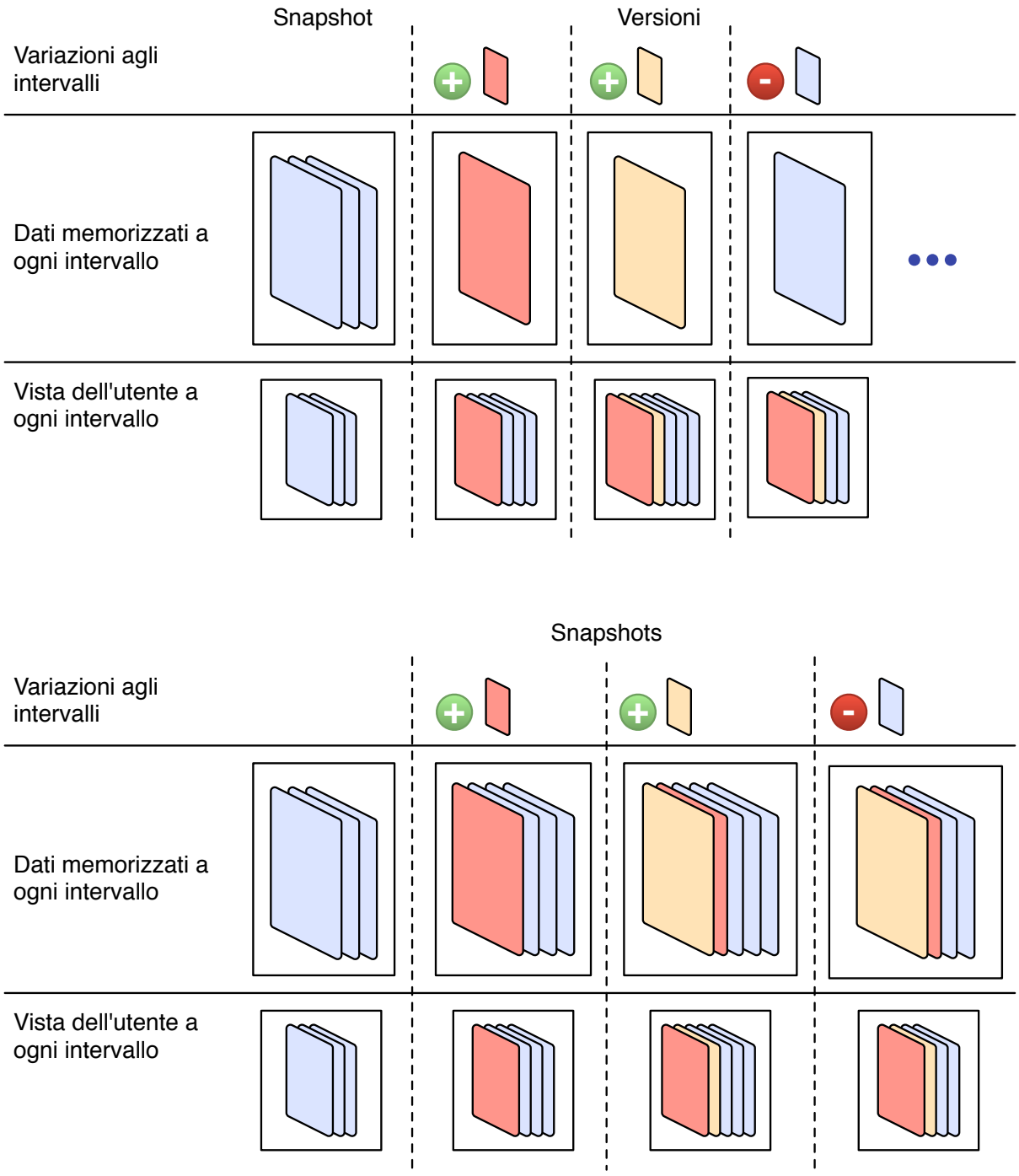
Il backup incrementale di presta al mantenimento di vaste informazioni storiche e occupa meno spazio, pur memorizzando tutte le versioni, compresi vecchi file cancellati. Tuttavia nella maggioranza di infrastrutture di backup incrementale è molto difficile se non impossibile ottenere i dati consistenti senza adottare il software utilizzato per eseguire il backup, il che aggiunge un fattore di rischio alle opportunità di recupero.

Il backup a snapshot utilizza uno snapshot per ogni versione del backup incrementale. Il sistema si basa sul copiare la totalità delle informazioni in ogni procedura di esecuzione. Questa scelta può essere implementata tramite sincronizzazione della struttura di backup con ogni servizio da cui estrarre informazioni o tramite la generazione di pacchetti da copiare. Il backup a snapshot offre una semplicità estrema di realizzazione dell'infrastruttura e di ripristino dei dati ma richiede di gestire esternamente il concetto di copie storiche.

La scelta del backup incrementale è utile per personal backup o backup di infrastrutture caratterizzate da molte macchine su ciascuna delle quali sono apportate modifiche ridotte tra un intervallo di esecuzione a l'altro e sulle quali vengono elaborate informazioni ricorrenti (come messaggi di posta elettronica ricevuti da tutti i computer in un ufficio). Strategie di backup a snapshot si adattano a infrastrutture in cui deve essere possibile ripristinare un servizio in pochi minuti e dove sono coinvolti o file moto piccoli e modificati molto frequentemente oppure file di grandi dimensioni, come interi filesystem. Nel progetto Newbackup è stata implementata un'infrastruttura di backup a snapshot.

[ii] Time Machine è una tecnologia per il personal backup e ripristino di file e metadati. La tecnologia sarà inclusa in Mac OS X a partire dalla release 10.5.0. Maggiori informazioni presso http://www.apple.com/it/macosx/leopard/timemachine.html

Figura 7.1: Confronto tra strategie di backup: incrementale e a snapshot



7.2.2. Strategie di replicazione

In una infrastruttura di backup completa devono essere considerate le opzioni di replicazione. La replicazione del backup è essenzialmente il backup del backup, ovvero l'applica-

zione di una strategia di backup al sistema principale. Tale strategia è soggetta alle stesse considerazioni espresse in precedenza. In prima battuta è necessario scegliere tra un sistema di backup attivo, cioè in grado di sostituire il primo backup o un sistema passivo, cioè una copia delle sole informazioni. Nelle infrastrutture di dimensioni maggiori è possibile sfruttare l'elevato controllo che si ha dell'informazione una volta presente nel sistema per applicare strategie di replicazione del solo storage, cioè della componente di stoccaggio e non della logica applicativa del sistema.

Il metodo di replicazione più semplice è senza dubbio generare uno storage alternativo che subisce modifiche imposte dall'infrastruttura principale. Si tratta semplicemente di applicare una ridondanza allo strumento di backup per prevenirne i guasti. Alternativamente è possibile realizzare una copia totale del sistema e renderla in grado di sostituire il sistema principale in ogni momento. In tal caso devono essere realizzate procedure di backup che coinvolgono i servizi di cui memorizzare i dati e procedure di gestione che competono al dialogo tra le due repliche e alla determinazione di quale delle due può attivare le funzioni necessarie al backup. Sarebbe infatti uno spreco attingere più di una volta alla stessa informazione da due sorgenti diverse in grado di comunicare tra loro.

La problematica principale nel replicare la logica applicativa totalmente sta nel dover gestire procedure automatizzate su dati che possono provenire da sorgenti delle quali non si ha il controllo. Per fare un esempio, in una semplice architettura dove esiste un master che preleva i dati e li replica su uno slave, è necessario applicare strumenti di verifica utili allo slave per accertarsi di disporre di informazioni consistenti, prima di considerarle tali in eventuali procedure. Questo rende la replicazione totale di sistemi incrementali un procedimento molto complesso.

La replicazione del solo storage appare l'espediente in grado di garantire la maggiore scalabilità, sia nel numero di repliche sia nelle opportunità di incrementare lo spazio disponibile per ciascuna replica. Naturalmente tale gestione è anche al più complessa e costosa.

Nel progetto newbackup è stato impiegato un sistema di replicazione totale tra due infrastrutture. Il software è stato realizzato per assolvere alle funzioni di master e di slave a se-

conda delle circostanze. Nella funzione normale il master si occupa del backup e replica i soli dati sullo slave. In caso di malfunzionamento o replica incompleta, lo slave prende il ruolo di master e si occupa del backup primario. Quando il vecchio master viene riattivato è possibile applicare una replica totale in senso inverso.

La replicazione, oltre che necessaria e utile, è anche richiesta per legge. Nel DPS è necessario certificare di disporre di un sistema di backup in sede remota, in grado di essere distaccato da eventuali incidenti anche di grave entità occorsi all'infrastruttura principale.

7.2.3. Sistema di copie storiche

Realizzare un sistema di copie storiche significa mantenere uno stato consistente delle informazioni per momenti nel tempo diversi dall'immediato. Un sistema di backup storico mantiene le informazioni passate in base alla data o a intervalli di acquisizione. Il vantaggio di mantenere uno storico è quello di disporre di informazioni passate a scopo di analisi o per ripristino in caso di errori commessi in momenti precedenti. I sistemi di backup incrementale realizzano uno storico memorizzando ogni versione come un differente istante consistente. I sistemi di backup a snapshot possono essere arricchiti da gestioni esterne delle copie storiche dettagliate.

Quando si tratta di backup storico è necessario stabilire l'età massima delle copie e ogni quanto mantenere una nuova versione. Entrambi i parametri impattano sulla stima dello spazio necessario la backup perché riguardano la vita del sistema nel tempo, addentrandosi in istema approssimative sulla crescita del volume dei dati. Una stima in favore di sicurezza si rende necessaria, sovradimensionando lo spazio concesso al sistema. Per il resto le scelte sono determinate dalla frequenza dei cambiamenti nelle informazioni dell'infrastruttura e dalla necessità di mantenere informazioni storiche.

Il progetto Newbackup è stato dimesionato per poter mantenere una tipologia di storico variabile, a seconda del servizio da coprire. E' possibile realizzare uno storico con cadenza:

- Settimanale, per informazioni che subiscono modifiche nel corso della settimana e sono raramente soggette a ripristino per errori di giorno in giorno;
- Mensile, per servizi come DNS che subiscono modifiche saltuarie;

- Giornaliero, per sistemi in cui è necessario mantenere informazioni storiche precise e consistenti;
- Ultimo mese, per sistemi soggetti a modifiche frequenti su dati con poca rilevanza sul lungo periodo.

La modalità ultimo mese non conserva dati del mese precedente, le altre modalità mantengono uno storico per data che può espandersi arbitrariamente nel tempo. In particolare, non esiste limite alla modalità giornaliera, per questo viene usata per memorizzare i log di navigazione del servizio Proxy, da mantenere per legge per almeno due anni.

Oltre alle modalità descritte, il sistema Newbackup mantiene sempre disponibili le informazioni della settimana corrente.

7.2.4. Pull backup e push backup

La sicurezza dei dati memorizzati in un sistema di backup è fortemente influenzata dalla scelta compiuta per il sistema di ottenimento dei backup. Le strategie possibili sono principalmente due: consentire al sistema di backup di prelevare i dati o renderlo passivo al deposito da parte di altri sistemi. Strategie differenti coinvolgono la negoziazione del backup installando applicativi client e server per instaurare la connessione classica nella direzione richiesta dall'applicativo (come avviene nel prodotto open source Bakula, per il backup su nastro).

Realizzare un sistema di backup pull, ovvero in grado di prelevare le informazioni, offre numerosi vantaggi. Anzitutto è possibile di trasferire l'intera logica di configurazione e organizzazione dei file direttamente sul sistema di backup, rendendo le sorgenti di informazioni del tutto passive al prelievo. Questa strategia migliora il controllo esercitato su tutte le procedure eseguite in automatico. Dal punto di vista della sicurezza, nell'ipotesi di isolare il sistema di backup in un'area apposita della rete locale, è consigliabile limitare il più possibile le credenziali di accesso all'area. Una strategia pull offre la possibilità di limitare le procedure di backup a connessioni instaurate dall'interno dell'area protetta, rimuovendo la necessità di controllo di accesso. Questa strategia costringe tuttavia a realizzare un accesso su ogni sorgente di informazione (macchina singola o porzione di rete) da dedicare al backup, pertanto è

necessario prendere precauzioni affinché questo accesso non possa essere sfruttato altrimenti. Il principale lato negativo di una strategia pull è la limitata capacità da parte del sistema di backup di determinare eventuali cambiamenti ai dati da memorizzare nel momento in cui avvengono. Questo limita le strategie di backup incrementale alla sola opzione ad intervalli di tempo. Per quanto riguarda i backup a snapshot non esistono limitazioni di alcun tipo.

I backup di tipo push sono caratterizzati da un minore controllo sul modo in cui i dati vengono trasferiti presso il sistema. Garantendo alle sorgenti l'accesso diretto alla destinazione, è possibile ricevere informazioni di nessuna rilevanza o addirittura dannose. Per realizzare un corretto sistema push è necessario formalizzare delle chiare politiche su cosa può essere mantenuto e come i dati devono pervenire. Deve inoltre essere approntato un sistema di controllo di accesso presso il backup in grado di escludere o limitare azioni da parte di client che non rispettano le policy formulate. Un'eccellente applicazione di un sistema a push è la realizzazione di una SAN, trattata nel paragrafo successivo.

Il sistema Newbackup utilizza un backup a pull, con accorgimenti per evitare lo sfruttamento delle credenziali di sistema di backup in modo maligno.

7.2.5. Pianificazione del backup

Gli utenti di un sistema di backup, intesi come fornitori delle informazioni da memorizzare possono essere applicazioni o servizi in esecuzione su server oppure utenti reali. Questa prima distinzione è determinante nelle scelte relative alla pianificazione dei backup. Una strategia di backup carente di pianificazione temporale rischia di ottenere informazioni non consistenti o di avere transazioni non corrette a causa dell'interferenza con processi attivi sulle macchine. E' possibile distinguere alcune possibili pianificazioni delle procedure di backup:

- Backup a tempo: ad intervalli di tempo definiti viene eseguito un backup totale o incrementale delle informazioni previste. Questa scelta si adatta bene alle situazioni in cui non sono necessari interventi di utenti reali per il corretto funzionamento della procedura. Una corretta pianificazione in questo senso deve accertarsi che le procedure di backup, generalmente pesanti nei confronti delle infrastrutture di rete, siano eseguite in momenti di

scarso utilizzo. La pianificazione temporale deve riflettersi anche sul comportamento di sistemi IDS installati;

- Backup su richiesta dell'utente: estremizzando è possibile paragonare questa procedura al collegamento di un disco rigido esterno per la copia dei dati. In una infrastruttura di backup centralizzata di tipo push il fornitore dei dati sceglie quando inviare informazioni al sistema di backup. In una infrastruttura di tipo pull è possibile aggiungere interfacce per consentire agli utenti di fare richiesta di procedure di backup immediate o pianificate per il futuro;
- Backup su evento: si tratta del tipo comportamento di sistemi a versione per il personal backup. Ogni volta che cambia qualcosa nelle informazioni o nei file da memorizzate viene registrato il cambiamento o la storia del cambiamento. Il backup è implicito nell'azione di modifica e consente di annullare i cambiamenti per la durata del periodo massimo di backup.

Il sistema Newbackup utilizza una strategia di backup a tempo, pianificando le operazioni di backup come singole o tutte insieme. Sono possibili anche soluzioni ibride, dove oltre a una pianificazione temporale si eseguono backup su richiesta (è il caso di BackupPC);

7.2.6. Dimensionamento di un sistema di backup

Il dimensionamento del sistema influenza numerose scelte tecniche si implementazione. Il sistema di backup deve garantire la necessaria scalabilità a operare per lungo tempo mantenendo le informazioni consistenti. Questo requisito si traduce in un vincolo di affidabilità dei componenti di stoccaggio e di flessibilità dello spazio utilizzato. E' necessario poter estendere lo spazio secondo necessità o riservare uno spazio di dimensioni tali da non richiedere estensioni all'interno del tempo di obsolescenza dell'hardware utilizzato. La scelta implementativa principale è dunque tra un sistema di backup semplice o quello che si potrebbe definire un backup "enterprise grade".

Contestualizzando la definizione al singolo computer, un backup semplice è un disco fisso esterno in cui l'utente ha cura di copiare periodicamente i dati importanti, mentre un backup enterprise grade è l'accesso a una rete specifica per la raccolta di dati centralizzata.

Estendendo la definizione alla realtà di una sala macchine di piccole dimensioni un backup semplice è una logica per gestire il trasferimento di informazioni verso l'equivalente di un disco esterno ma con la sicurezza e la dimensione necessaria alla situazione. Il backup enterprise grade diventa una struttura con una logica centralizzata che si appoggia sulla rete di raccolta e trasferisce le informazioni a unità di storage che offrono un elevato grado di consistenza interna e una semplice scalabilità nel numero. E' immediato notare come le due scelte possano determinare i costi dell'infrastruttura.

Il sistema Newbackup utilizza una strategia di backup semplice. Per realizzarlo sono stati acquistati due sistemi di storage gemelli, nei quali è stato ricavato lo spazio necessario sovrastimando a tempi superiori all'obsolescenza dei dischi mantenendo architetture ad alta affidabilità. I due sistemi ospitano spazio disco, logica di prelievo e logica di gestione. Inoltre sui dispositivi stessi è stata implementata la logica per la replicazione dei dati. Le strategia di backup descritte si applicano soltanto a informazioni prodotte dai servizi forniti dal CeDoc e non alle singole piattaforme in uso alle biblioteche. Il software di backup, vero scopo del progetto, è tuttavia in grado di essere decentralizzato qualora le biblioteche che dispongono di collegamenti adeguati ne facciano richiesta.

E' stato stimato che una strategia di backup enterprise grade avrebbe avuto un costo solo per l'hardware di due o tre volte superiore a quello sostenuto per l'intero progetto, software incluso. A scopo di confronto si riporta tuttavia un esempio realistico di strategia di backup enterprise grade. La strategia si basa sulla realizzazione di un sistema SAN e un sistema NAS.

SAN è l'acronimo per Storage Area Network: si tratta di una rete parallela a quella utilizzata per connettere numerosi computer con il solo scopo di veicolare dati per il backup o lo storage in generale. Una SAN può vivere parallelamente alla rete locale (LAN) oppure sfruttarne gli stessi dispositivi. Un sistema SAN completo utilizza un controller che gestisce lo spazio di archiviazione, determinando chi può accedere all'archivio e attraverso quali protocolli. I singoli utenti del backup, siano essi server o singoli computer, fanno riferimento a uno spazio disco remoto attraverso la SAN sul quale sono autorizzati a eseguire i backup. E' possibile pianificare i backup o consentire agli utenti di attivarli manualmente. Al di là dello spazio disco attraverso la SAN, gli utenti del backup non hanno nozione.

Il sistema si completa con uno o più NAS, ovvero Network Attached Storage. Si tratta di unità di archiviazione collegate direttamente alla rete o gestite da un controller. Il vantaggio dell'utilizzare tali dispositivi è che si può affidare al controller l'intera logica di gestione delle informazioni assieme al software necessario per gestire lo spazio dinamicamente, in modo da poter aggiungere parti al NAS in modo trasparente ai livelli precedenti. Il sistemi NAS fanno affidamento su protocolli come iSCSI[iii] e su tecnologie come RAID[iv] per garantire la necessaria affidabilità del sistema di trasferimento e di memorizzazione. E' immediato notare come un'architettura NAS+SAN offra scalabilità, affidabilità ed efficienza applicate a un sistema di backup in grado di offrire i suoi servizi per un parco macchine molto vasto. Questa è la principale differenza tra un sistema di backup ad hoc e un sistema enterprise grade.

Un esempio di sistema enterprise grade usa il software open source OpenFiler[v] per fare da SAN controller e strumentazioni proprietarie per gestire i NAS che offrono il reale spazio su disco. La scalabilità è garantita dal supporto dei sistemi linux (RedHat Enterprise Linux[vi]) per la tecnologia LVM (Logical Volume), in grado di virtualizzare lo spazio disco a livello software e mascherare come unico contenitore un gruppo di volumi la cui consistenza è gestita a basso livello.

7.3. Caratteristiche dell'infrastruttura da coprire

7.3.1. Esclusioni

Il sistema Newbackup è stato progettato come strumento dedicato ai soli servizi fondamentali. La copertura dei personal computer in uso al personale non è stata inclusa tra le specifiche del sistema, tuttavia è auspicabile l'introduzione di un SAN controller per la centraliz-

---

[iii] iSCSI è un layer di trasporto standardizzato per l'utilizzo della tecnologia SCSI-3 su reti TCP/IP

[iv] Reduntant Array of Indipendent Disks (RAID) è una tecnologia che consente di utilizzare più dischi raggruppati in array logici. Ogni array viene visto dal sistema operativo come un disco unico e può applicare politiche interne di aumento delle prestazioni (striping) o aumento dell'affidabilità (mirroring)

[v] OpenFiler è una piattaforma affermata per la gestione dello storage enterprise. Il prodotto è Open Source e pensato per operare su sistemi RedHat Linux. maggiori informazioni presso http://www.openfiler.com/

[vi] RHEL è la distribuzione di linux realizzata da RedHat inc. dedicata al mercato enterprise. Il prodotto è Open Source ma a pagamento. Spesso si utilizza la versione completamente gratuita di RHEL, priva del supporto tecnico ufficiale ma altrettanto funzionale. Maggiori informazioni presso http://www.redhat.com/rhel/ e http://www.centos.org/

zazione dei dati provenienti da tali terminali, il cui spazio di memorizzazione sarà gestito dal sistema Newbackup.

In nessun caso è prevista la copia di dati provenienti da postazioni in uso alle biblioteche; tali procedure non sono attuabili, in considerazione di due caratteristiche fondamentali:

- L'infrastruttura di rete non è in grado di supportare trasferimenti di grandi quantità di dati (backup) su rete wan senza inficiare la normale operatività dei collegamenti. Questo problema può essere aggirato con procedure di backup notturno in condizioni di operatività normale (ovvero non nel caso di copie di emergenza);
- La decentralizzazione dei sistemi rende l'assistenza tecnica molto complessa, pertanto casi di ripristino dei dati su rete wan non possono essere seguiti con la cura necessaria;
- La maggior parte dei sistemi non offre dati da memorizzare. I casi di perdita di informazioni a causa di guasti su dischi fissi è molto ridotta. I computer vengono prevalentemente utilizzati per la navigazione in internet e la ricezione di posta elettronica. Per i dati relativi a quest'ultima, un sistema di memorizzazione efficace è garantito dal buffer del server di posta, impostato a una settimana. Per le biblioteche che desiderano un backup di documenti in genere si offre un disco esterno da utilizzare personalmente. Per infrastrutture in grado di gestire un server di backup locale è tuttavia accessibile l'intera infrastruttura Newbackup, con copia in sede remota.

7.3.2. Zone della rete

Esclusi i personal computer rimangono soltanto le macchine che ospitano i servizi gestiti presso il CeDoc. Queste macchine si trovano in un ambiente generalmente controllato, suddiviso i alcune aree principali:

- DMZ: area che ospita tutti i servizi accessibili dall'esterno della rete del CeDoc, come il sito Web e strumenti correlati. Quest'area è caratterizzata dall'isolamento rispetto all'interno della rete, a causa dell'elevata esposizione agli accessi dall'esterno. Per questa zona della rete è imperativo adottare una strategia di backup pull;

- Inside: area più vasta della rete che ospita i personal computer. I servizi ospitati in quest'area sono ad uso interno con requisiti di sicurezza ridotti. L'area inside non è considerata sicura, pertanto si devono applicare strategie valide per la DMZ;
- Serverfarm: area che ospita servizi a uso interno e ad alto grado di criticità. L'accesso in quest'area dall'esterno è gestito da policy severe. Il sistema di backup risiede in quest'area e ha accesso diretto a tutti i servizi, per quanto consentito dalle politiche di accesso alle singole macchine.

In relazione alla struttura della rete appare utile la realizzazione di un sistema di backup pull, il quale dall'interno di un'area protetta ottiene i dati accedendo a servizi che si trovano nell'area stessa o che si trovano all'esterno ma sono comunque soggetti ad accessi di vario tipo. Poiché le informazioni devono essere veicolate all'interno di un'ara protetta è necessario garantire l'assoluta riservatezza del canale di trasmissione e la resistenza degli accessi ad attacchi esterni.

Per la politica di sincronizzazione esiste una seconda unità di backup in sede remota. Questa unità è raggiunta attraverso alcuni instradamenti che connettono la LAN del CeDoc alla LAN dell'infrastruttura remota. La LAN del CeDoc è in grado di garantire la riservatezza nel tratto che separa la zona serverfarm al tratto di connessione in fibra ottica con la LAN remota. La LAN remota non è attualmente in grado di garantire la riservatezza del tratto finale, per questo motivo è consentita la sincronizzazione soltanto tramite canali cifrati e sicuri.

7.3.3. Sistemi operativi interessati

In generale è possibile applicare una prima distinzione tra i servizi interessati da procedure di backup. L'infrastruttura del CeDoc è caratterizzata da:

- Macchine Unix in prevalenza. Si tratta di macchine che ospitano servizi su sistemi operativi FreeBSD e OpenBSD. Alcune macchine utilizzano sistemi operativi basati su kernel Linux. Macchine Unix e Linux sono soggette alle stesse politiche di backup con strumenti comuni. Per la trasmissione di dati cifrati tra il sistema di backup (su piattaforma FreeBSD) e questi servizi è possibile utilizzare canali SSH;

- Macchine Windows in minoranza. Per la manutenzione dei sistemi Microsoft Windows e per la gestione di servizi esterni al CeDoc sono state utilizzate macchine che ospitano sistema operativo Windows 2003 Server. Queste macchine possono offrire dati da memorizzare attraverso il sistema di condivisione documenti in uso su sistemi Microsoft: SMB. Questo sistema offre metodi di autenticazione e gestione della sessione, tuttavia non raggiunge lo stesso grado di sicurezza ottenibile attraverso SSH;
- Backup secondario. L'unità di backup in sede remota utilizza FreeBSD, pertanto la sincronizzazione può avvenire con le procedure note per sistemi Unix.

L'infrastruttura è costituita da sistemi che utilizzano metodi di trasferimento dati consolidati e accessibili, pertanto è opportuno che lo strumento di backup basi il suo funzionamento su tali metodi. La scelta riduce la complessità di realizzazione di un sistema di backup autonomo e sfrutta lo sviluppo eseguito da terze parti per disporre di un sistema di trasferimento all'avanguardia.

7.3.4. Natura dei dati

In merito alle tipologie di dati da memorizzare, è possibile suddividere l'infrastruttura in due tipi di servizi:

- Servizi che producono una ridotta quantità di informazione, generalmente limitata alla configurazione della macchina. Questi servizi hanno un backup con storico ridotto o assente; la memorizzazione dei file di configurazione ha lo scopo di consentire un ripristino rapido del servizio in caso di guasto della macchina che lo ospita. Questo tipo di backup non è presente in infrastrutture che utilizzano cluster per la virtualizzazione, dove i servizi vengono rimpiazzati rapidamente attraverso snapshot di macchine virtuali;
- Servizi che producono grandi quantità di dati, tipicamente database, log o spazio disco. Questi servizi sono caratterizzati da una quantità ridotta di dati necessari per operazioni di ripristino e da una quantità in proporzione molto elevata di dati a valore informativo, sulla cui dimensione è possibile esercitare solo un controllo ridotto. Queste infrastrutture affrontano procedure di backup con uno storico generalmente lungo nel tempo.

Le due categorie devono essere soggette a procedure di backup indipendentemente dalla loro struttura e quantità di dati da memorizzare. Questo vincolo si traduce in alcuni requisiti fondamentali per il sistema di backup:

- Il sistema deve essere dimensionato per una quantità di dati crescente, in relazione alle proiezioni di aumenti formulate per il parco macchine da coprire. Una strategia di backup enterprise grade può essere adeguata, tuttavia è possibile effettuare il dimensionamento anche tramite mezzi più economici;
- Il sistema deve essere flessibile in termini di backup e prevedere procedure di backup adatte sia a ridotte quantità di dati che al trasferimento di file di grandi dimensioni. Deve inoltre essere possibile adottare procedure differenti per la singola macchina, in modo da modellare tutti i servizi su un solo servizio tipo, il quale ha una parte di configurazione e può avere una parte di dati da trasferire separatamente;
- La flessibilità in backup deve riflettersi sulle procedure di storico, prevedendo la possibilità di applicare procedure differenti su una singola macchina. Questo consente di modellare uno storico a lunghezza differente a seconda del tipo di informazione, benché riferito allo stesso servizio.

7.3.5. Vincoli alla pianificazione

La maggior parte dei sistemi del CeDoc esegue continuamente operazioni automatizzate. Le macchine Windows non hanno procedure pianificate, ma quasi tutti i sistemi Unix utilizzati devono sincronizzare numerose procedure interne o compiute assieme ad altre macchine. In particolare esistono alcune tipologie di operazioni che possono interferire con le procedure di backup:

- Riavvii. Alcune macchine sono soggette a riavvii automatici. Questi eventi sono di durata non sempre predicibile (check casuali del disco rigido, per esempio) e non possono avvenire durante l'operazione di backup. E' opportuno pianificare i backup prima dei riavvii, dimensionando i tempi in favore di sicurezza per prevedere possibili operazioni di backup prolungate;

- Database dump. Trasferimenti di interi database o di parti di essi possono impattare sulla consistenza delle informazioni memorizzate se esse comprendono anche i dati del database o dei file di destinazione. Per includere il backup nelle procedure di dump è opportuno eseguire una preparazione al backup (si veda in seguito) subito dopo aver operato il dump e averne accertato il termine;
- Produzione continua di dati. La produzione continua di dati può interferire con la correttezza delle informazioni memorizzate. In generale le operazioni di backup coinvolgono la lettura, tuttavia è preferibile che i file in corso di prelievo non siano soggetti ad altre procedure. Nei sistemi Unix, sistemi come newsyslog[vii] gestiscono i file di log ciclici, principali produttori di informazioni in continua evoluzione.

Il problema principale legato alla pianificazione è dovuto alla difficoltà, in un sistema pull, di determinare il momento adatto per eseguire il backup. Gli stretti intervalli di sincronizzazione possono essere allargati dividendo il backup in due fasi:

- Preparazione al backup. I file da memorizzare vengono preparati per il trasferimento. I file possono essere compressi in un unico archivio per ridurre i tempi di trasferimento. La preparazione può avvenire in un momento differente al trasferimento, favorendo la sincronizzazione. La fase di preparazione può essere pianificata in modo da evitare dump o produzione continua di dati.
- Trasferimento (backup vero e proprio): trasferimento dei file da memorizzare. La fase di trasferimento può avvenire anche contemporaneamente a dump o produzione continua di dati.

7.4. Struttura del sistema di backup

7.4.1. Funzionalità

In base alle caratteristiche dall'infrastruttura da coprire e dalle possibili scelte accennate per la realizzazione di un sistema di backup, è possibile delineare le caratteristiche principali

[vii] Newsyslog è uno strumento configurabile per la gestione e l'archivio di file di log. Maggiori informazioni presso http://www.weird.com/~woods/projects/newsyslog.html

del sistema Newbackup, corredate dalle indicazioni sull'implementazione. Il sistema Newbackup deve fornire:

- Un sistema di backup, ovvero prelievo dei dati rilevanti da qualsiasi macchina che monti sistema Windows o Linux/Unix. Il metodo di prelievo deve essere adatto sia a file di grandi dimensioni sia a grandi quantità di file piccoli. La preparazione dei file deve essere coordinata con le normali attività delle macchine in modo da non interferire con l'operatività del CED e deve proporre metodologie diverse a seconda della criticità del prelievo. Il prelievo avviene con strategia a snapshot a intervalli di tempo definiti;
- Un sistema di copie storiche, in modo da mantenere backup che spaziano in periodi di tempo molto vasti con intervalli arbitrari. Le copie storiche avvengono con cadenza mensile, settimanale, possono mantenere tutto l'ultimo mese oppure mantenere uno storico giornaliero di lunghezza arbitraria;
- Un sistema di replicazione in grado di propagare i backup su uno o più sistemi "slave" collocati in sedi differenti. Il sistema di replicazione è attivato a tempo e deve consentire in ogni momento alla replica di attivarsi come sistema principale, pur limitando la quantità di dati trasferiti in condizioni di normale operatività;
- Un modulo di verifica da eseguire periodicamente per accertarsi dell'integrità dei file di acquisizione più recente;
- Un sistema integrato di logging, per mantenere traccia di ogni azione intrapresa dagli altri sistemi allo scopo di analisi del comportamento dell'infrastruttura, sia in situazioni operative standard sia in caso di anomalie;
- Un sistema di notifiche in grado di rilevare anomalie nei processi operativi standard e informare via e-mail dell'anomalia corredando la notifica di informazioni aggiuntive;
- La possibilità di configurazione illimitata, ovvero l'opportunità per un operatore che conosce il sistema di configurare efficacemente le procedure di backup ed eventualmente intervenire sull'elenco stesso delle azioni da compiere.
- La facilità di deployment su strutture remote. Il sistema deve essere di immediata installazione e deve consentire la propagazione di modifiche anche significative su tutte le installazioni facenti capo a una singola infrastruttura.

Le specifiche formulate sono fornite da un sistema software che può essere definito un complesso e funzionale strumento per la pianificazione di procedure di backup. Il software è realizzato attraverso alcuni script per la shell di Unix bash[viii] i quali sfruttano molte tecnologie ancora una volta patrimonio del mondo Unix, quali SSH[ix], rsync[x], tar[xi] e samba[xii]. Gli aspetti tecnologici di ciascuno strumento saranno trattati parallelamente alla descrizione del suo comportamento all'interno del sistema.

7.4.2. Moduli

Il sistema è suddiviso in moduli, ciascuno dei quali ha una specifica funzione. Sebbene il software sia stato realizzato attraverso un linguaggio di scripting, è stata adottata una struttura tipica dei linguaggi a oggetti per favorire la manutenzione e la chiarezza sulle funzioni dei moduli.

Le seguenti descrizioni fanno riferimento ai file di configurazione intendendo il file di configurazione principale e il file di configurazione per ciascun profilo, dove ogni profilo definisce i parametri di una singola procedura di backup. Possono essere definiti profili illimitati che fanno riferimento a una singola macchina.

I moduli realizzati per il sistema sono i seguenti:

main

Modulo principale che offre i servizi di avvio del software. L'interazione con il sistema avviene sempre soltanto attraverso il modulo main. L'utente deve lanciare il modulo specificando la posizione del file di configurazione principale e quale procedura attivare tra quelle

---

[viii] Bourne Again Shell (BAS) è una shell compatibile con sh realizzata dalla Free Software Foundation per il sistema operativo GNU. Bash integra elementi di Korn Shell (ksh) e C Shell (csh). Sorgenti e documentazione presso http://directory.fsf.org/bash.html

[ix] SSH (Secure Shell) è uno strumento per la trasmissione di dati in modo sicuro su una connessione TCP/IP. Maggiori informazioni presso http://www.ssh.org/

[x] Rsync è una utility Open Source che fornisce un trasferimento di file veloce e incrementale. Maggiori informazioni presso http://rsync.samba.org/

[xi] GNU tar è un sistema di archiviazione che gestisce numerosi formati. Maggiori informazioni presso http://directory.fsf.org/tar.html

[xii] Samba è un software Open Source che fornisce servizi di stampa e condivisione di file su protocollo SMB/CIFS senza l'adozione di sistemi operativi Windows. Sorgenti, binari e documentazione disponibili presso http://www.samba.org/

disponibili. La sintassi operativa di main è allineata con la tipica sintassi per gli strumenti a linea di comando Unix.

deploy

Modulo indipendente al servizio degli sviluppatori del sistema. Deploy legge la configurazione e determina tutte le macchine impostate come possibili bersagli delle procedure di sincronizzazione. Su ciascuna, offre la possibilità di propagare separatamente soltanto la logica applicativa. Deploy può essere utilizzato per propagare modifiche fatte al software su tutte le postazioni attive in produzione.

parser

Il modulo viene lanciato da main e si occupa della verifica dei file di configurazione. parser esegue i file di configurazione e verifica che ciascuna impostazione sia coerente con le specifiche del sistema. Infine, parser compie alcune verifiche sulla coerenza dello stato del sistema in merito ai backup storici disabilitati automaticamente per malfunzionamenti di processi precedenti.

director

Il modulo gestisce l'esecuzione dei backup pianificati. Main richiama director fornendogli le opzioni scelte dall'utente per la procedura di backup. Director determina se eseguire backup di solo alcune macchine oppure dell'intero parco macchine e per ogni configurazione determina le modalità di backup; infine esegue il modulo backup adeguato all'intervento su ciascuna configurazione. Il modulo director attende lo stato di uscita di backup e disattiva la procedura di backup storice per macchine per cui l'esito non è stato positivo.

backup-rsync

Il modulo è chiamato direttamente da director e può essere usato per configurazioni di backup di macchine Unix. Il modulo backup-rsync legge la configurazione ottenuta e attiva la procedura di trasferimento attraverso la sincronizzazione con la sorgente. Al termine, legge lo stato di uscita della procedura e lo notifica sia via email, allegando il log, sia a director per ulteriori provvedimenti.

backup-tar

Questo modulo le stesse funzioni del modulo backup-rsync ma con un trasferimento di file compressi attraverso SSH, senza sincronizzazioni. Il modulo può essere utilizzato per il backup di macchine Unix.

backup-smb

Questo modulo esegue le stesse funzioni del modulo backup ma attraverso il protocollo smb. Il modulo smb è utilizzato per il backup di macchine Windows.

history

Come director è il gestore dei backup attuali, history è il gestore dei backup storici. history viene chiamato da main per l'intera procedura di backup storico e determina quali profili sono soggetti alla procedura e quali sono stati disabilitati per emergenza. Per tutti i profili attivi chiama poi il modulo compact che esegue il backup storico.

compact

La funzione di compact è determinare i file più recenti per ciascun profilo e operare una copia in base alle politiche di backup storico imposte da history. Compact si occupa anche della notifica di errori e ignora i profili per cui la configurazione non è valida per la data corrente (ad esempio backup settimanali richiesti nel giorno sbagliato).

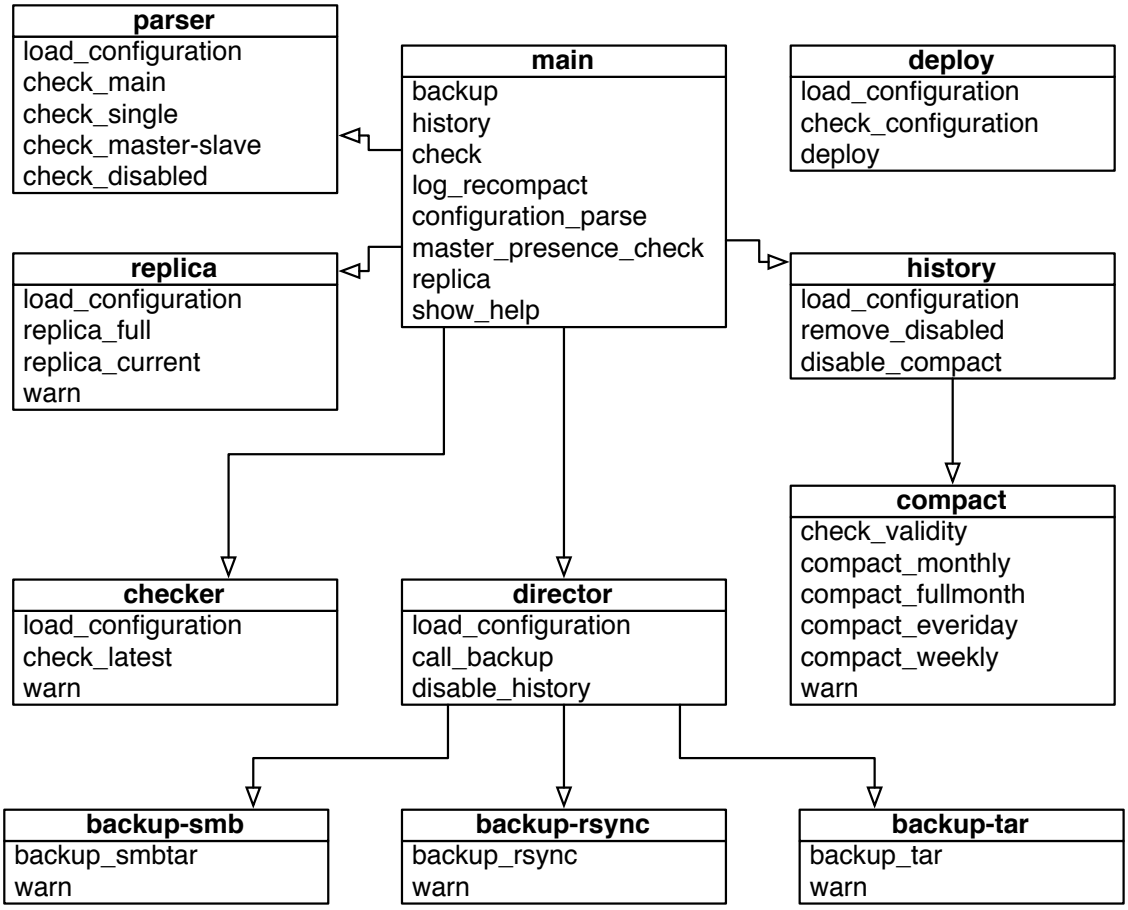
replica

Il modulo replica è chiamato direttamente da main e si occupa di gestire la replica su un numero arbitrario di postazioni remote. Per ogni postazione viene letta la configurazione che specifica la destinazione e la modalità di replica. Il modulo si occupa di gestire la replica e della notifica di errori.

checker

Chiamato direttamente da main, il modulo checker esegue la verifica dei file più recenti del backup storico di ciascun profilo. Il modulo si occupa della notifica di errori di verifica.

Figura 7.2: Diagramma dei moduli del sistema Newbackup



7.5. Procedure

Le procedure sono le normali sequenze che regolano le funzioni del sistema. Ogni sequenza coinvolge più moduli del sistema e più funzioni di software esterni.

7.5.1. Backup

La procedura di backup è la condizione di funzionamento fondamentale espressa dal sistema. In breve si tratta di scegliere a quali profili applicare l'operazione di trasferimento ed effettuare il trasferimento stesso. La procedura di backup può utilizzare tre differenti metodi per il trasferimento dei file dai servizi sorgenti: rsync over SSH, tar over SSH, tar over SMB. I primi due metodi sono utilizzati per il backup di macchine Unix, il terzo per il backup di macchine Windows.

rsync over SSH

Il ruolo di rsync è sincronizzare due percorsi, siano essi appartenenti a uno solo spazio nomi oppure facenti capo a posizioni distanti attraverso la rete. Se il trasferimento avviene tra percorsi sulla stessa macchina, rsync utilizza i normali metodi di spostamento o copia. Se il trasferimento è attraverso la rete, rsync può utilizzare un protocollo apposito (chiamato rsync) o appoggiarsi su altri protocolli di trasferimento, in questo caso SSH.

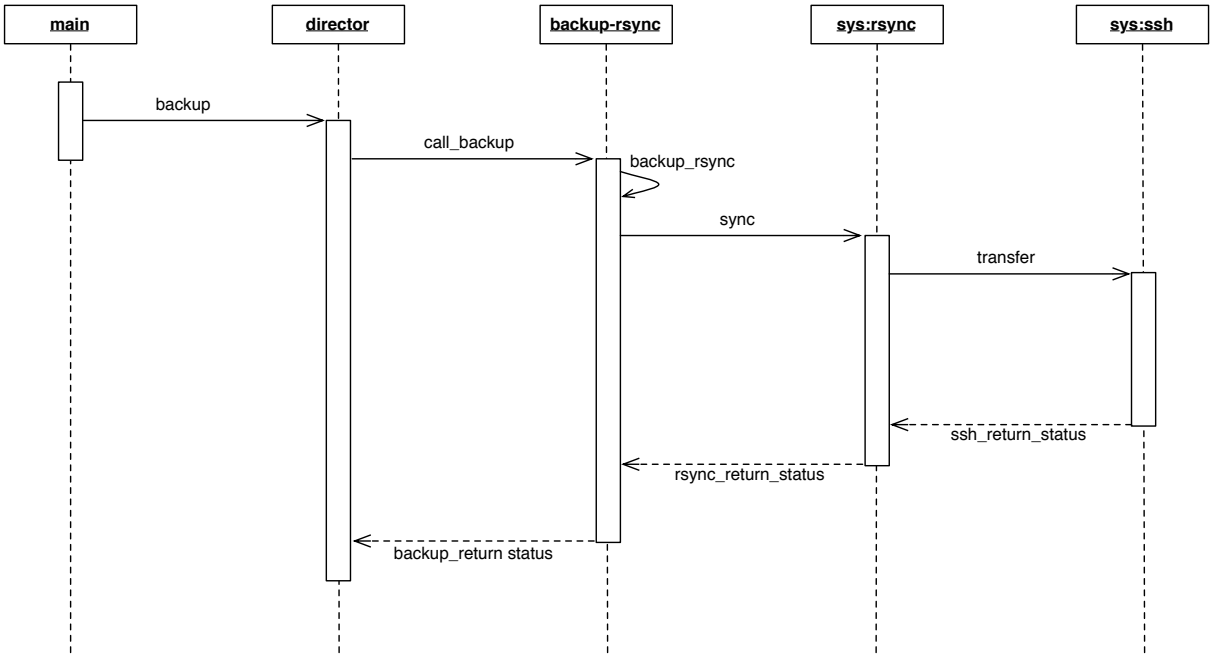
SSH offre un sistema di autenticazione in chiave pubblica basato su certificati (o chiavi) ed eventualmente inserimenti di password (quest'ultima caratteristica non viene utilizzata nel sistema) per due nodi su una rete. Una volta eseguite l'autenticazione e l'autorizzazione, i due endpoint scambiano informazioni attraverso un canale di trasmissione cifrato in chiave simmetrica. In generale la scelta sacrifica le prestazioni in favore di un sistema di trasmissione sicuro e riservato. Il sottosistema SSH è presente nella maggior parte delle distribuzioni Linux e nelle varianti di Unix, pertanto appare il metodo ideale su cui basare il trasferimento sicuro dei file.

Indipendentemente dal metodo di trasferimento, rsync è in grado di determinare quali file nel percorso sorgente non sono presenti nel percorso di destinazione o differiscono da controparti presenti. Costruita la lista, le modifiche vengono propagate alla destinazione. Esistono numerose opzioni per la gestione del trasferimento e del percorso di destinazione; nel sistema Newbackup l'utente può utilizzare una sintassi arbitraria per rsync, specificata attraverso la configurazione dei profili.

L'architettura del sistema richiede una sincronizzazione da svolgersi specificando un percorso remoto da cui prelevare i dati. Per consentire il corretto funzionamento delle copie storiche, i dati remoti devono essere preparati. Per preparazione del backup si intende l'esecuzione sulla macchina remota di una procedura che preleva dati da punti sensibili e crea un archivio per il trasferimento. In genere si utilizza una destinazione sola per gli archivi, i quali hanno nel nome il nome della macchina e il giorno della settimana. In questo modo ogni esecuzione produce un archivio che sostituisce quello eseguito lo stesso giorno della settimana precedente. Il percorso contenente sette archivi viene interamente sincronizzato nella procedura di backup, per questo l'unità minima per il backup con rsync over SSH è la settimana. E'

possibile configurare la procedura di preparazione per produrre archivi con storico mensile, tuttavia la procedura è svolta più efficacemente dal sistema di backup.

Figura 7.3: Sequence diagram della procedura di backup con rsync



Nel grafico le classi con intestazione sys rappresentano funzioni di sistema o fornite da applicazioni installate su di esso. Ogni azione intrapresa nelle procedure è accompagnata da una scrittura sul file di log. Tali registrazioni non sono riportate negli schemi.

tar over SSH

La procedura di backup via tar sfrutta un'altra potenzialità dell'implementazione open source di SSH, la possibilità di eseguire un comando sulla macchina e trasferire l'output attraverso il canale cifrato. Il canale SSH vviene generato, poi sulla macchina di destinazione viene lanciato il comando tar, in grado di creare un archivio compresso di un numero arbitrario di oggetti. Nella fattispecie si utilizza l'implementazione di gnutar per realizzare un archivio con l'algoritmo gzip. L'output del comando non viene ridiretto su file come in una semplice archiviazione ma viene trasferito attraverso il canale SSH, al termine del quale viene ridiretto su file. In questo modo, date una sorgente e una destinazione, la destinazione instaura

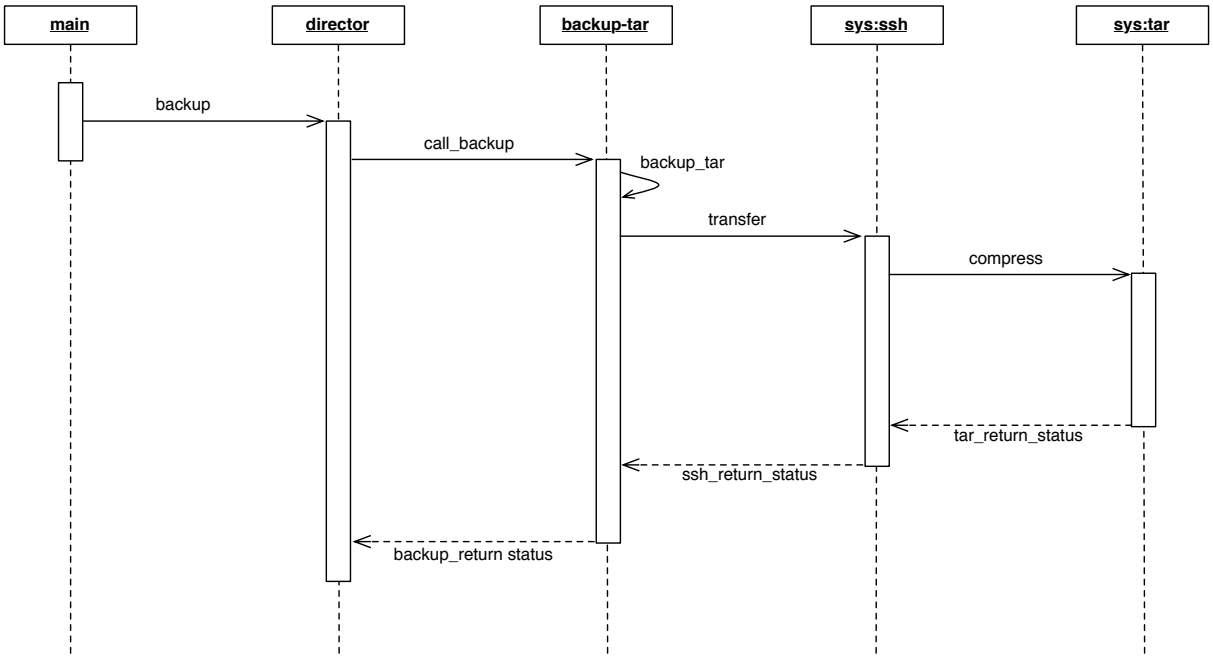
un canale SSH con la sorgente, preleva i dati compressi e li inserisce in un file una volta a destinazione.

Questa scelta non necessita di alcun tipo di preparazione e non richiede lo spazio di archiviazione disponibile per la settimana corrente sulla machina fornitrice. Il sistema a tar over SSH viene utilizzato prevalentemente per apparati di rete o piccole macchine con un sistema in sola lettura senza disco fisso.

Sebbene sia più efficiente nella gestione interna, il backup via tar ha alcuni limiti rispetto a rsync. Con rsync è possibile disacoppiare il momento della preparazione da quello del trasferimento, risolvendo i problemi di sincronizzazione con servizi interni della macchina; tar deve prelevare i dati nel momento in cui avviene il collegamento. Inoltre, rsync offre un eccellente sistema di notifica e controllo della connessione, assolutamente assente nella soluzione tar. Infine, l'architettura a rsync consente a un utente ad alti privilegi (anche root) di creare l'archivio per poi farlo trasferire con credenziali di accesso più basse. Questa opportunità consente a un utente a bassi privilegi di prelevare un backup contenente file normalmente non accessibili (su cui non ha diritto di lettura). La procedura di backup via tar richiede di poter leggere attraverso l'utente remoto tutti i dati che si desidera memorizzare. Questa scelta esclude alcuni file critici dal backup, a meno di non consentire un accesso automatizzato alle sorgenti come utente root, procedura altamente sconsigliata.

Questo sistema non dovrebbe essere utilizzato per il trasferimento di grandi quantità di file perché non ha alcun controllo di flusso o di riuscita dell'operazione di compressione sull'endpoint remoto. In altre parole, se il comando tar interrompe l'output inaspettatamente non esiste un controllo rapido ed efficiente sul sistema di backup che può segnalare un insuccesso. L'archivio deve essere verificato interamente per determinare se non esiste il terminatore che ne segnala il completamento.

Figura 7.4: Sequence diagram della procedura di backup con tar

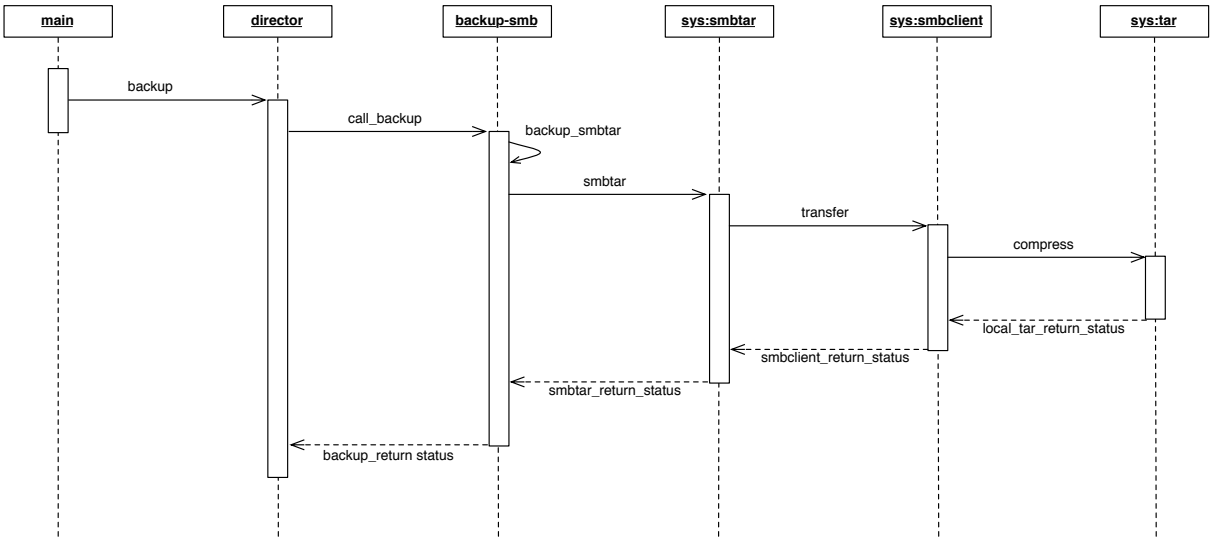


smbtar

Per il backup di macchine Windows non c'è molta scelta di metodi se non l'utilizzo del protocollo SMB. A tal fine si utilizza l'implementazione open source del protocollo ideato da Microsoft: samba. Il pacchetto software samba-client offre alcuni strumenti utili per accedere in vari modi a condivisioni smb remote. Tra di essi è presente lo script smbtar, il quale si appoggia sul programma smbclient per raggiungere una condivisione, autenticarsi e trasferire i dati. smbtar veicola i dati trasferiti attraverso tar, per ottenere un archivio dell'interno contenuto della condivisione.

Il sistema di backup a smbtar può essere utilizzato efficacemente con la stessa strategia di tar over SSH, con la differenza del canale di trasmissione. Le procedure relative al backup con smbtar sono svolte dal modulo smb.

Figura 7.5: Sequence diagram della procedura di backup con smbtar



7.5.2. Backup storico

La procedura di backup storico non richiede interazione con le sorgenti delle informazioni. In sede di backup storico si considerano valide le informazioni disponibili presso la settimana corrente di ciascuna macchina come punto tdi partenza della procedura.

Ad ogni esecuzione il backup storico preleva il file più recente dalla pool settimanale di ogni macchina e lo copia nelle pool storiche configurate, rinominandolo opportunamente.

Per ogni profilo sono possibili più opzioni di backup storico, che possono essere attivate indipendentemente l'una dall'altra:

- Backup settimanale: nel giorno della settimana specificato il sistema preleva l'archivio designato e lo trasferisce nella pool settimanale dello stesso profilo. Il file viene rinominato apponendo la data del giorno in forma seriale;
- Backup mensile: il backup mensile opera esattamente come quello settimanale ma a un giorno del mese specifico, anch'esso da configurare. Il file copiato viene rinominato apponendo la data del giorno in forma seriale;

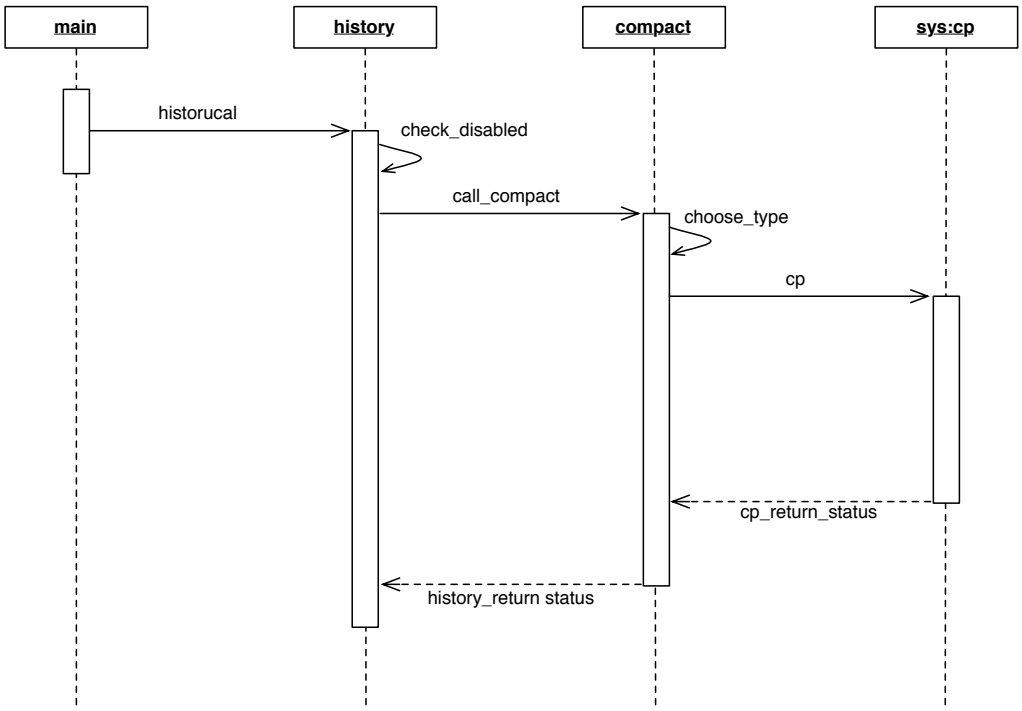
- Backup dell'ultimo mese: ogni giorno il file più recente viene copiato nella pool per l'ultimo mese e rinominato in accordo al giorno del mese. Il mese successivo nello stesso giorno il file verrà sovrascritto;
- Backup giornaliero: ogni giorno il file designato viene copiato nella pool per lo storico giornaliero e rinominato apponendo la data del giorno.

Le procedure di backup storico ignorano modalità non configurate per i profili, perciò ogni opzione di storico deve essere esplicitata per attivare il backup. I backup storici ignorano inoltre i giorni per cui non sono configurati i backup. Se si desidera effettuare un backup settimanale, ad esempio, è necessario specificare il giorno della settimana in cui eseguirlo o accettare il valore predefinito.

Il backup storico è gestito dal modulo history, il quale accede a ogni file di configurazione disponibile. Per ogni profilo configurato preleva le opzioni di backup e lancia un backup storico per ogni differente modalità configurata. Il backup vero e proprio viene eseguito dal modulo compact, il quale è passivo rispetto alla rilevazione delle configurazioni. Il modulo compact automatizza soltanto al procedura di spostamento e modifica del nome del file, senza distinzione per il profilo.

Prima di lanciare qualsiasi procedura di backup storico, il modulo history verifica che la macchina su cui insiste il profilo non sia nella lista di macchine disattivate. Se la macchina è disattivata il backup storico potrebbe compromettere la consistenza di storici precedenti (si pensi al caso di backup dell'utlimo mese).

Figura 7.6: Sequence diagram del backup storico

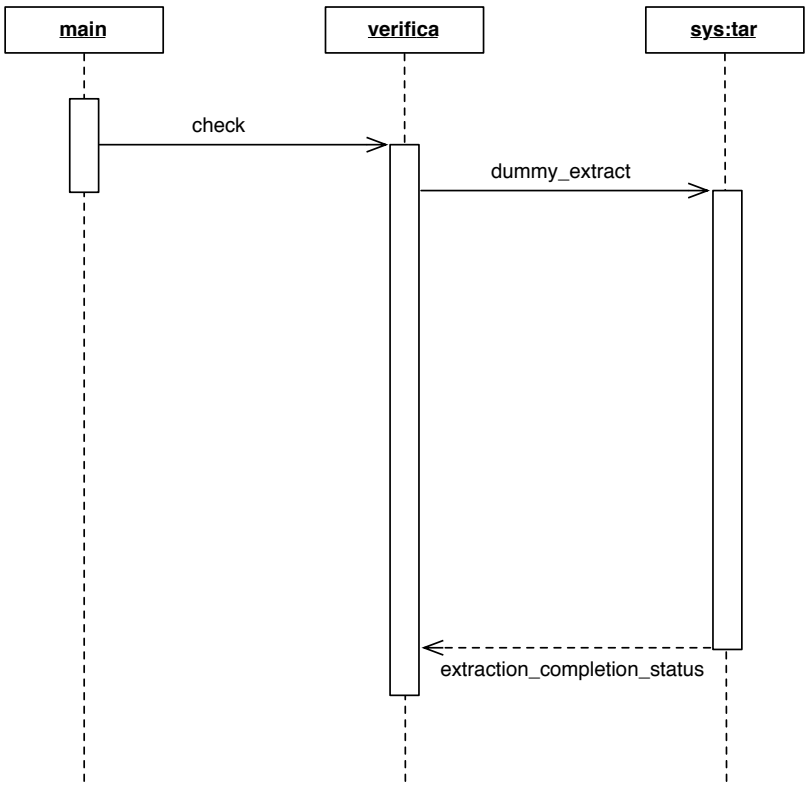


7.5.3. Verifica delle pool

La pool di backup storico possono essere verificate periodicamente. Le operazioni di verifica devono essere attivate esplicitamente, come quelle di backup e di storico. Il sistema utilizza il modulo checker per determinare le directory in cui possono esistere pool di backup per i profili attivi. Per ogni pool viene raggiunto il file inserito più di recente e si attiva la procedura di verifica.

La verifica attualmente consiste in una prova di ricostruzione del contenuto dell'archivio. Se la ricostruzione è possibile fino al termine dell'archivio, il file si considera consistente. In caso di inconsistenza l'amministratore viene avvertito. La verifica utilizza l'applicativo di sistema tar.

Figura 7.7: Activity diagram della procedura di verifica



7.5.4. Replicazione

La procedura di replicazione si svolge sempre tra coppie di sistemi: una è considerata master e una è considerata slave. Esistono due possibili procedure di replicazione, una replicazione per copia attiva e una per ripristino. La replicazione per copia attiva compete al normale funzionamento di una infrastruttura di backup in sede remota: giornalmente i dati della pool corrente vengono sincronizzati dal master allo slave. La procedura di ripristino serve a riattivare l'intero sistema dopo un guasto. La replica per ripristino copia l'intero sottoalbero a cui fa capo il sistema di backup, comprese le pool di backup storico.

Nel normale funzionamento del sistema ogni volta che viene lanciata una procedura deve essere specificato se eseguirla in modalità master o in modalità slave. La modalità viene confrontata con lo stato dell'applicazione e la procedura si attiva solo se esiste una corrispondenza. Alcune procedure possono modificare lo stato dell'applicazione; in questo modo è possibile definire un set di procedure da eseguire se in modalità master e un set da eseguire se in

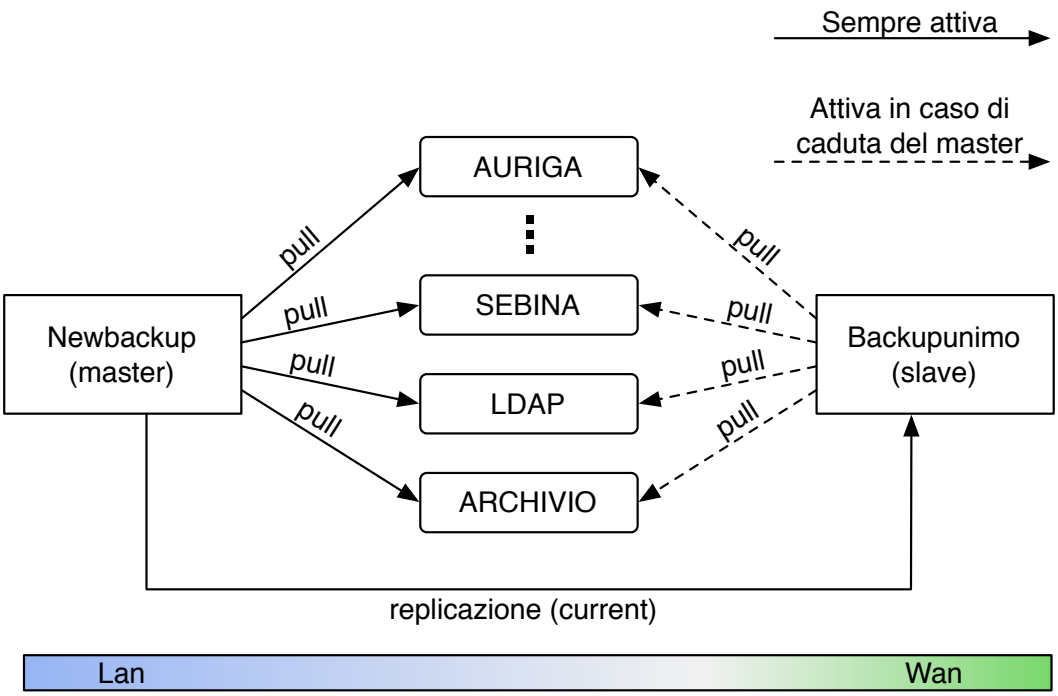
modalità slave. Nella fattispecie è possibile avere un sistema di backup che esegue il prelievo dei file, poi una replica in modalità copia attiva su un sistema slave. Il sistema slave può essere configurato per eseguire il prelievo dei dati solo in modalità master ed essere impostato per operare in modalità slave. Se la sincronizzazione fallisce il sistema slave cambia il suo stato e sostituisce il sistema master fino a un intervento tecnico.

La normale procedura di replica utilizza rsync su SSH per copiare soltanto le modifiche sul sistema slave. Al termine della sincronizzazione viene trasferito un ultimo file detto success mark. Questo file ha lo scopo di notificare allo slave che la procedura di replica è andata a buon fine. Lo slave deve controllare l'esistenza di questo file prima di tentare qualsiasi procedura di backup storico. La verifica di presenza del success mark è la differenza tra eseguire la procedura di backup storico in modalità slave piuttosto che in modalità master.

Il sistema Newbackup consente la presenza di più di uno slave associato a un solo master. Ogni slave può avere caratteristiche diverse rispondenti a diverse specifiche della procedura di replica. Inoltre un sistema cosiderato slave di un altro può essere a sua volta master per un qualsiasi numero di ulteriori sistemi. Questa architettura consente di organizzare la struttura delle repliche in un'architettura ad albero per propagazione di aree designate su macchine o sottoalberi differenti.

L'architettura operativa del sistema Newbackup consiste di un master localizzato al CeDoc e di uno slave posizionato presso il DSIT, Università degli Studi di Modena e Reggio Emilia.

Figura 7.8: Schema della replicazione nel sistema Newbackup



7.5.5. Configurazione

Il sistema newbackup dispone di due percorsi di configurazione: la configurazione principale del software e la configurazione dei profili. Entrambi sono nella forma di script che vengono eseguiti nel contesto della shell corrente con il comando source. Gli script contengono soltanto assegnamenti di variabili, le quali portano le direttive di configurazione.

Configurazione principale

Il file di configurazione principale (predefinito main.conf) contiene direttive generali per l'applicazione. Attraverso questo file è possibile configurare:

- Utente che l'applicazione utilizza per tutte le operazioni. Questo utente deve coincidere con quello di sistema che lancia i processi, altrimenti nessuna operazione avrà luogo;
- Percorso principale dell'applicazione. Il percorso principale è generalmente utilizzato come base per tutti gli altri percorsi. Nella configurazione predefinita, se si decide di rispettare la disposizione di base dell'applicazione, è sufficiente impostare il percorso principale e gli altri vengono configurati automaticamente;

- Percorso degli script. Percorso della directory che contiene gli script che implementano i moduli applicativi;
- Percorso per i log. I log contenenti notifiche e note sull'esecuzione del software si trovano nel percorso log. Questo percorso contiene anche la lista di profili disabilitati dal backup storico e le notifiche di errori;
- Percorso per le e-mail. Ogni volta che viene generato un messaggio di notifica via e-mail il corpo del messaggio viene salvato in questa directory;
- Percorso per i file. Le pool principali della settimana corrente vengono memorizzate nel percorso files;
- Percorso storico. Le pool per le quattro modalità di storico vengono memorizzate nel percorso storico. Questo percorso dovrebbe risiedere su un filesystem separato, se il software viene utilizzato per backup di grandi dimensioni;
- Percorso per i file di configurazione. nel percorso per i file di configurazione sono specificati i file di configurazione per i profili, un file per ciascuno. Ogni volta che il sistema fa riferimento a tutti i profili cerca tutti i file che terminano con .conf presenti in questo percorso. La procedura di backup può fare riferimento a profili specifici inserendo il nome relativo del file.
- Numero massimo di procedure di backup da eseguire contemporaneamente. Il numero predefinito è cinque per non sovraccaricare la rete ma può essere aumentato arbitrariamente.
- Posizione del comando per inviare messaggi. Newbackup utilizza sendmail[xiii] per veicolare i messaggi di posta, è necessario specificare il percorso del comando per l'invio. Questa scelta consente di specificare il binario in modo manuale o accettare la configurazione predefinita, funzionante sulla maggioranza del sistemi Unix e Linux.
- Destinatario per i messaggi amministrativi. Gli errori di competenza globale del sistema sono segnalati a questo indirizzo.
- Posizione del file di modalità. Questa direttiva specifica il percorso completo del file che contiene la modalità operativa configurata tra master e slave;

[xiii] Sendmail è un mail transfer agent Open Source incluso in molte distribuzioni Linux e varianti di Unix. Spesso viene utilizzato per generare messaggio di posta da una macchina senza utilizzare altri server SMTP. In contesti come la notifica di errori è particolarmente utilizzato. Maggiori informazioni presso http://www.sendmail.org/

- Nome del master. Se il sistema opera come slave deve conoscere il proprio master per poter operare verifiche periodiche di attività della controparte. Se una verifica fallisce, il sistema slave attiva l'operatività come master. Se il sistema opera come master la direttiva viene ignorata.
- Dati sulle repliche. Per ciascuno slave su cui effettuare repliche viene specificata una configurazione contenente il nome (o l'indirizzo ip),l'utente con cui eseguire la replica (attivo sul sistema remoto), il tipo di replica (tra current e full) e il percorso remoto su cui replicare.

Ciascun profilo viene configurato in un file separato, nel quale vengono specificate le direttive specifiche del profilo:

- Nome della macchina. Il nome viene utilizzato per fare riferimento alla macchina nelle procedure di backup. Nelle procedure di backup via SMB deve essere utilizzato il nome completo della macchina, non il nome netbios;
- Percorso remoto. Nel caso di backup via rsync il percorso è quello sulla macchina sorgente di una directory da cui prelevare il contenuto. nel caso di backup via SMB il percorso è il nome dello share sulla macchina sorgente. Nel caso di backup via tar il percorso è un insieme di path che fanno riferimento a file da includere nell'archivio;
- Directory di destinazione. Ogni profilo può avere una directory specifica nella pool. In questa directory viene memorizzata la settimana corrente. Lo stesso nome è utilizzato nel percorso dei backup storici per immagazzinare i dati delle quattro modalità;
- Specifica del comando rsync. Se si desidera utilizzare la modalità di backup con rsync è possibile personalizzare per intero il comando con i flag preferiti;
- Modalità di trasferimento. Questa direttiva può essere smb per i backup via SMB, tar per i backup con tar over SSH e la stringa per l'uso di ssh con rsync per i backup rsync (ad esempio --rsh="/usr/bin/ssh");
- E-mail del gestore del profilo. E' possibile specificare un indirizzo diverso da quello amministrativo per consegnare i messaggi che competono al solo profilo;

- Modalità di backup. E' necessario specificare quali modalità di backup adottare tra weekly, monthly, fullmonth, everyday;
- Giorno del mese e della settimana. Specifica dei giorni in cui eseguire rispettivamente il backup mensile e settimanale. Se non è specificato alcun giorno si utilizza un valore predefinito;
- Utente e password per il collegamento a uno share SMB. Utili soltanto per backup su SMB.

Il modulo parser è utilizzato per verificare la correttezza dei file di configurazione. Lanciando la funzione di parsing viene verificato il file principale e tutti i file dei profili.

7.6. Estensioni al sistema Newbackup

Il progetto Newbackup è in costante sviluppo per mantenere un sistema adeguato alle caratteristiche dell'infrastruttura del CeDoc. Periodicamente vengono aggiunte funzionalità nuove e migliorate quelle presenti. Si riportano alcune tra le caratteristiche più utili in programma per l'implementazione nei prossimi mesi.

7.6.1. Gestione unificata dei log e dei messaggi di avviso

Attualmente ogni modulo del sistema integra l'intero apparato di notifica degli errori e l'intera gestione dei log. Queste due funzioni si rivelano particolarmente utili per identificare le cause non ovvie di malfunzionamenti nella procedura di backup. Per cause non ovvie si intendono eventualità in cui la procedura non è fallita per problemi alla sorgente delle informazioni ma per anomalie di trasferimento o pianificazione.

La gestione attuale di log e notifiche, sebbene funzionante, è frammentaria e caratterizzata da una grande ridondanza del codice per l'implementazione. Una possibile estensione è l'accentramento dei servizi di notifica in un unico modulo adeguato al compito. Il modulo potrebbe essere utilizzato da tutti gli altri come server per la gestione dei messaggi, in particolare potrebbe:

- Gestire la messaggistica, distinguendo i messaggi in base alla destinazione e al tipo. In questo modo sarebbe possibile suddividere i messaggi tra messaggi di avviso e di errore oppure tra messaggi da destinare a un log e messaggi da spedire a un utente. Il modulo dovrebbe integrare la logica per propagare i messaggi dove richiesto;
- Occuparsi della rotazione dei log, utilizzando strumenti come newsyslog o una procedura di compressione ad hoc, sviluppata per il sistema newbackup;
- Costituire un elemento di pulizia del codice integrando le linee ridondati in un unico modulo.

7.6.2. Gestione dei metadati

Attualmente il sistema effettua la verifica dei dati archiviati tentando di decomprimere gli archivi. Questo approccio è sufficiente per la tipologia di archivi attualmente utilizzata che può essere decompressa utilizzando lo strumento tar.

La procedura di backup con rsync è limitata da questa scelta nel tipo di archivi che può recepire, le altre due procedure archiviano dati arbitrari e creano gli archivi direttamente sul software di backup. Estendere il numero di archivi possibili potrebbe offrire opportunità più vaste di backup su infrastrutture eterogenee.

L'estensione del supporto a più archivi richiede una modifica sostanziale alla gestione del nome dei file, della verifica dei dati e della sequenzailità dei backup. Per implementare queste modifiche può essere utilizzato un sistema di gestione dei metadati, in cui ogni pool di backup dispone di una sovrastruttura che può essere letta dal software di backup. Questa sovrastruttura è costituita da un insieme di file che memorizzano lo stato di ogni pool. Per ciascun file di backup nella pool viene memorizzato il nome, l'estensione e il chacksum[xiv]. Attraverso i metadati è possibile delegare la verifica a un semplice confronto del chacksum e determinare immediatamente le caratteristiche di una pool da cui estrarre i file per procedure come il backup storico.

[xiv] Tipicamente un numero o una stringa ottenuti dall'elaborazione byte a byte del file attraverso un algoritmo di hash. L'algoritmo dovrebbe produrre un checksum di dimensioni notevolmente inferiori al file e possibilmente costanti. Esso inoltre dovrebbe assicurare che due file differenti anche solo di pochi byte non producano checksum equivalenti. In questo modo, ricalcolando il chechsum in un secondo momento è possibile verificare univocamente se il file ha subito modifiche, evidenziate da un valore di ritorno differente.

la gestione dei metadati non è semplice utilizzando lo shell scripting. Per raggiungere il livello richiesto è possibile utilizzare un'applicazione esterna per gestire file semistrutturati (XML o altro) oppure realizzare un file di metadati semplice che può essere letto usando espressioni regolari.

7.6.3. Servizio di backup alle biblioteche

L'aumento di strutture bibliotecarie equipaggiate con collegamenti a banda larga rende possibile l'estensione del sistema di backup a strutture distribuite su rete wan.

In una ipotetica infrastruttura di backup remoto la biblioteca collegata a banda larga potrebbe disporre una macchina sulla quale verrebbe installata una versione del software di backup. La versione remota potrebbe utilizzare tecniche come il backup su tar over SSH o via smb per prelevare i dati dalle macchine in biblioteca. Il software potrebbe in tal modo adattarsi a infrastrutture che adottano sistemi Windows e Linux.

In seguito, il software remoto può propagare i suoi dati verso il backup centrale oppure può essere fornitore di dati per il backup centrale. Sul backup centrale verrebbero applicate soluzioni di storico personalizzate.

Questa architettura può estendere il sistema di backup trasformandolo in un servizio distribuito al servizio delle biblioteche.

CLASS DIAGRAM DELL'INTERFACCIA WEB

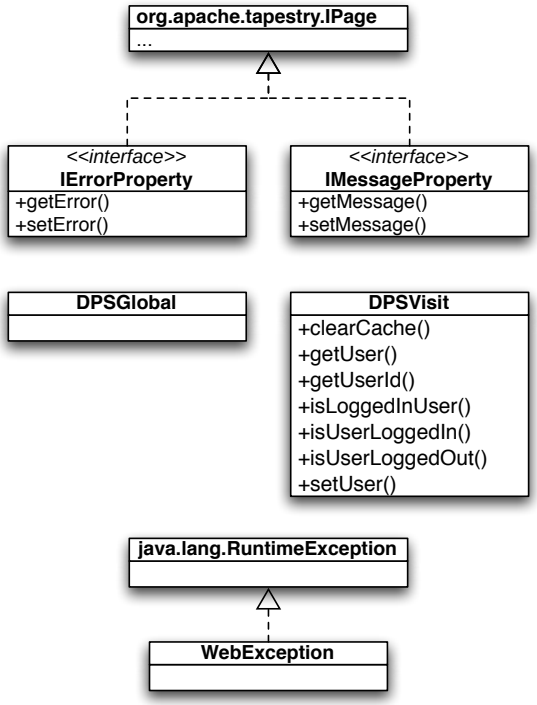
Si riporta in Class Diagram completo dell'interfaccia web, suddiviso per packages. Il class diagram riporta soltanto il contenuto del package dps.web, ulteriori dettagli sono coperti da diritto d'autore e non possono essere divulgati. I package presenti sono:

- dps.web
- dps.web.components
- dps.web.delegate
- dps.web.pages
- dpw.web.pages.op
- dps.web.services
- dps.web.services.impl
- dps.web.validators

dps.web

Package principale che racchiude tutte le classi dell'interfaccia web. A questo package appartengono alcune classi per l'implementazione dell'oggetto Global e l'oggetto Visit e altro. Questo package contiene tutti gli altri.

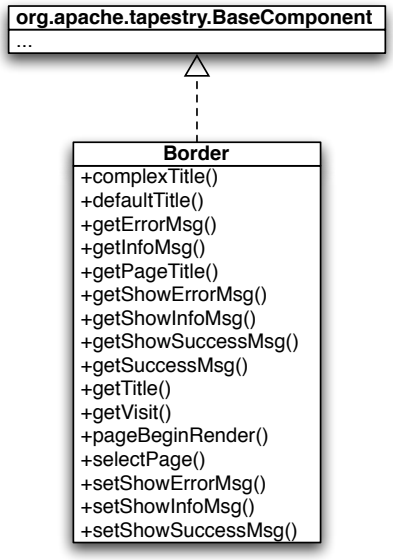
Class diagram per il package dps.web



dps.web.components

L'unico componente custom realizzato è Border:

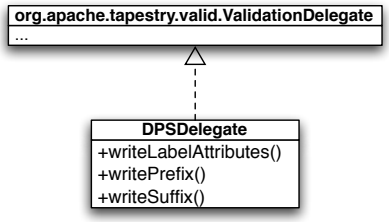
Class diagram per il package dps.web.components



dps.web.delegate

Il package contiene il delegate per la gestione dei messaggi di errore.

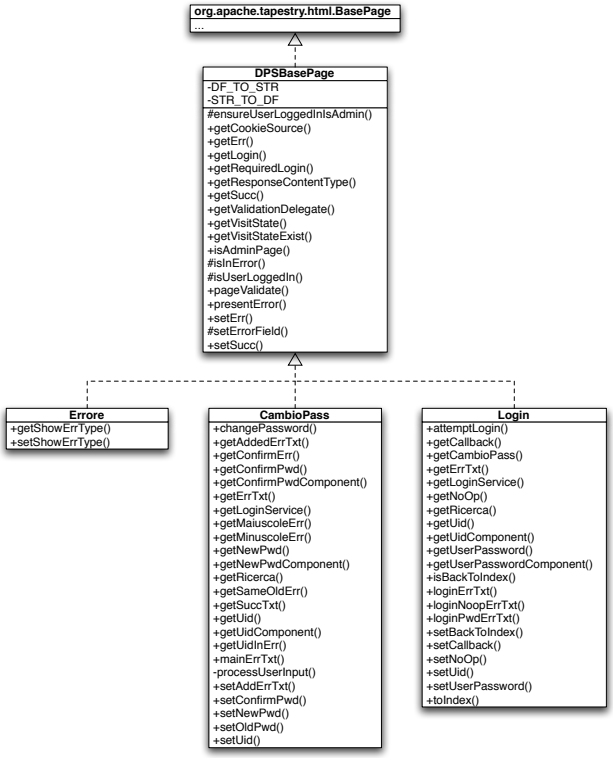
Class diagram per il package dps.web.delegate



dps.web.pages

Le pagine dell'Interfaccia Web appartengono tutte a questo package. Esiste un package innestato per le sole pagine protette da autenticazione, chiamato dps.web.pages.op. Le pagine di accesso libero sono Errore, Login e CambioPass. La pagina Start non ha una controparte Java.

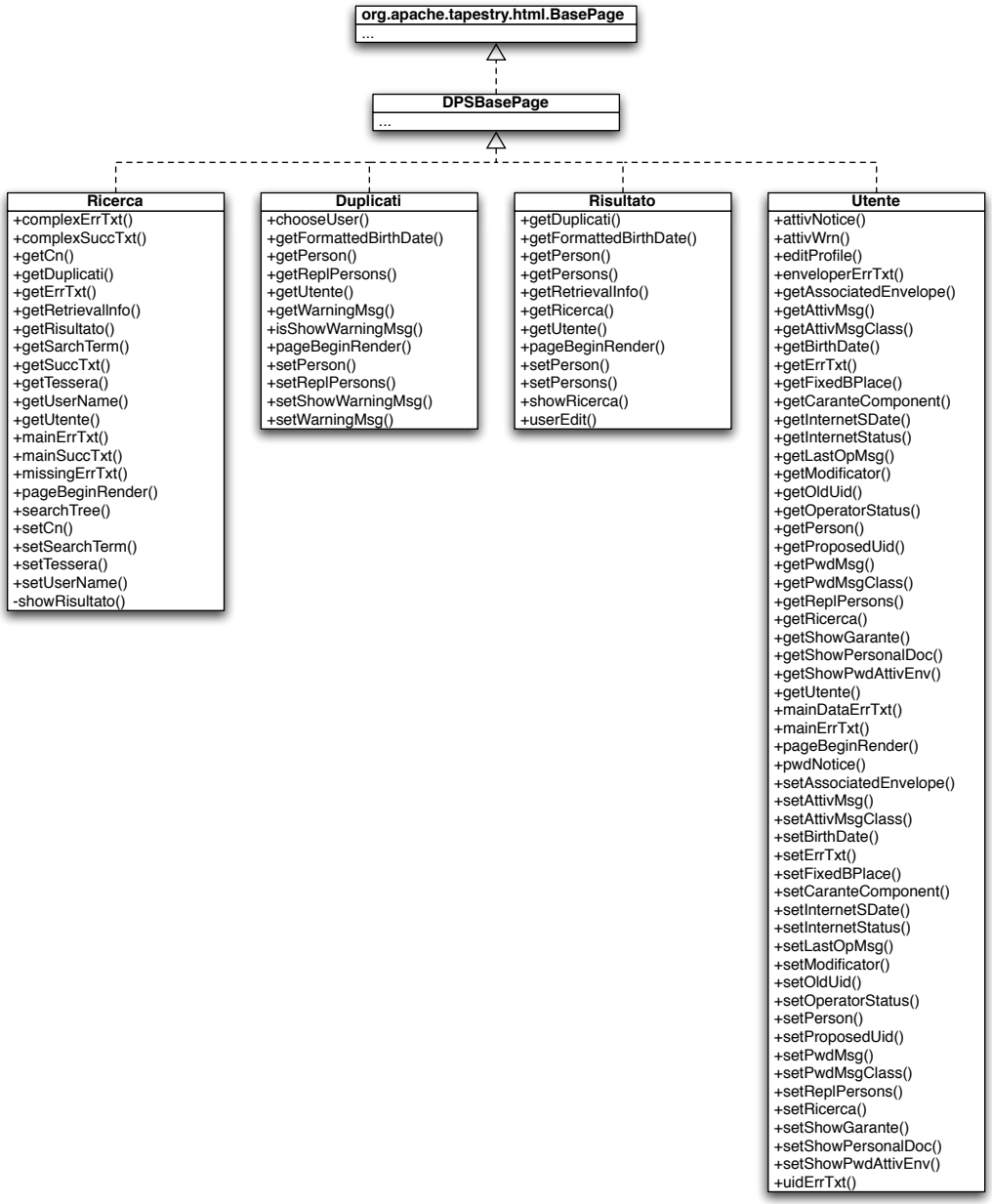
Class diagram per il package dps.web.pages



dps.web.pages.op

Le pagine protette da autenticazione fanno parte di questo package. Le pagine di op ereditano da DPSBasePage, la classe non è riportata per intero perché presente nel diagramma precedente.

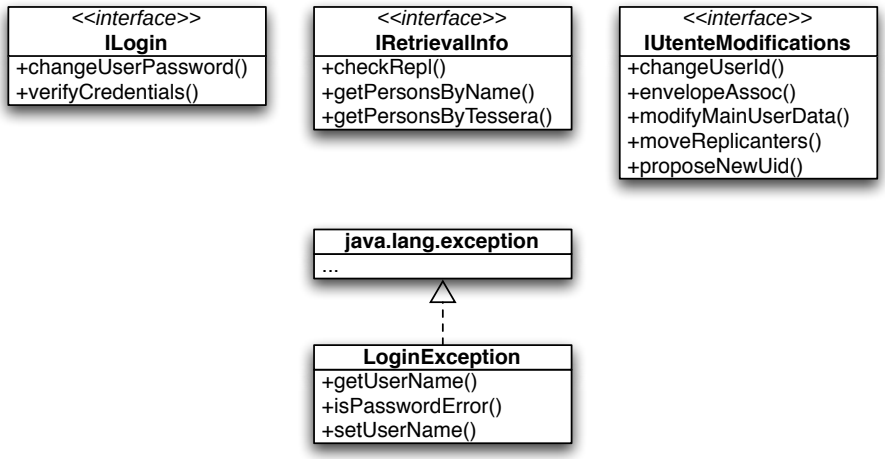
Class diagram per il package dps.web.pages.op



dps.web.services

I servizi utilizzati per l'interazione con LDAP e altre procedure sono definiti da interfacce in questo package. Il package successivo riporta l'implementazione dei servizi.

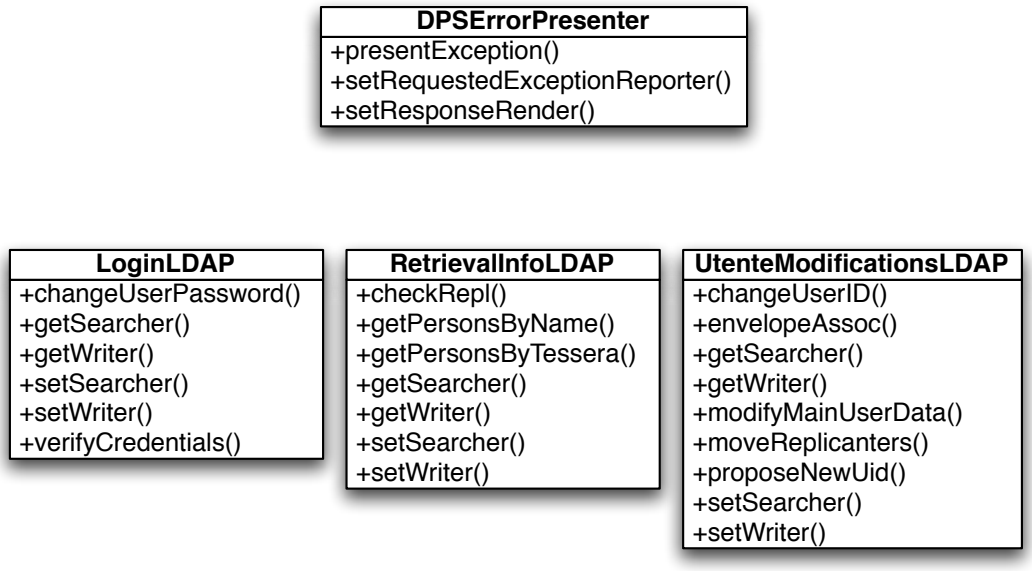
Class diagram per il package dps.web.services



dps.web.services.impl

Il package contiene le implementazioni delle interfacce ai servizi.

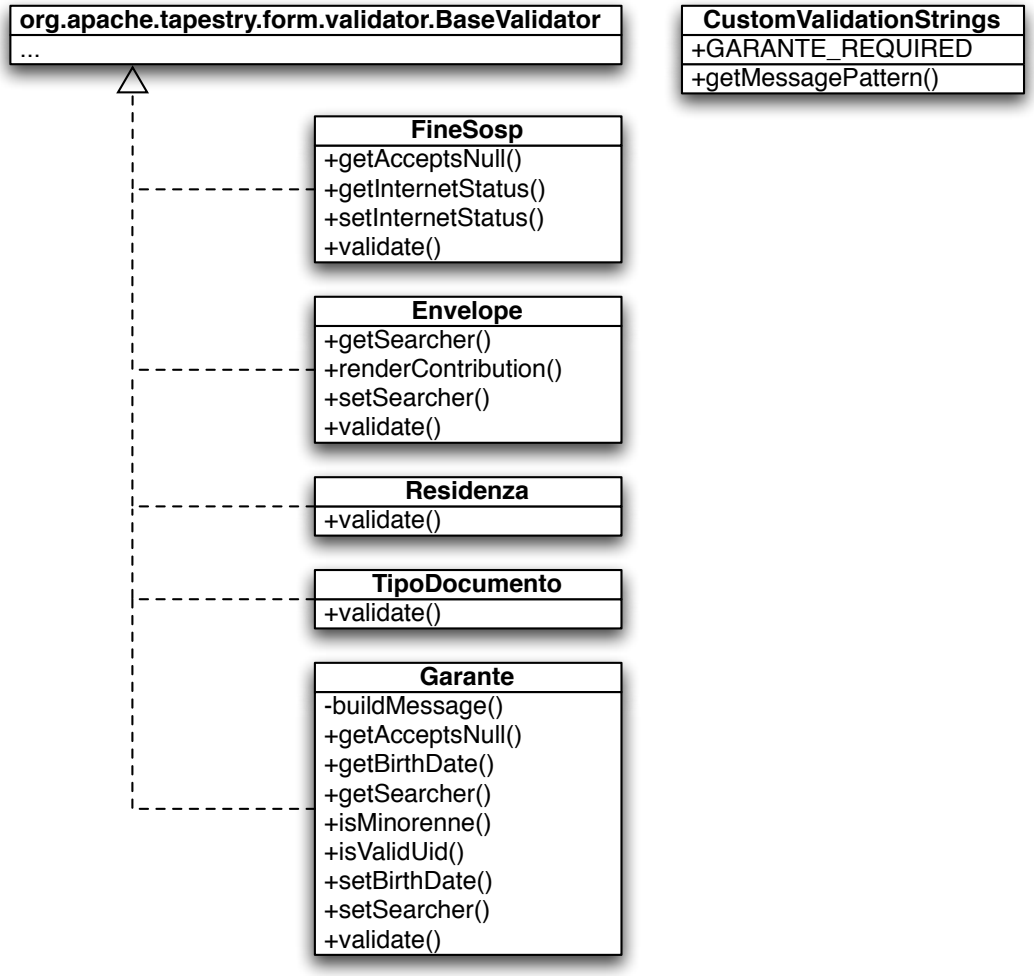
Class diagram per il package dps.web.services.impl



dps.web.validators

Package per l'implementazione dei custom validators.

Class diagram per il package dps.web.validatros



STRUMENTI SOFTWARE DI SUPPORTO AL PROGETTO

Nel corso della progettazione e realizzazione del progetto sono statu utilizzati numerosi strumenti Open Source per la programmazione e gestione del lavoro collaborativo. Questa sezione ha lo scopo di introdurre la funzione dei più rilevanti.

Controllo versione

Progetti come quello trattato in questo documento sono spesso caratterizzati da un processo di programmazione frammentario e svolto da più persone. Per garantire la consistenza del codice e la facile reperibilità di modifiche eseguite nel tempo si può adottare uno strumento di controllo versione.

Strumenti di questo tipo sono generalmente centralizzati e dispongono di un server nel quale viene memorizzato il codice delle varie versioni del software. Il punto in cui viene memorizzato il codice è detto repository. Quando il repository viene inizializzato è popolato con la prima versione del codice. Ogni successiva operazione di aggiornamento, detta commit, produce una nuova versione della quale vengono memorizzate solo le modifiche rispetto alla precedente. Le precedenti versioni non vengono mai modificate dal commit di una versione nuova.

Più programmatori possono lavorare sullo stesso codice: all'inizio del processo lavorativo il programmatore esegue l'operazione di checkout, ovvero l'aggiornamento della versione sulla propria macchina fino all'ultima disponibile. Alla fine del lavoro o a seguito di una modifica significativa si esegue il commit, portando la versione centralizzata al pari delle ultime modifiche. Se un commit si rivela dannoso per il codice è possibile eseguire il checkout di una versione precedente, o switch.

Normalmente i sistemi di controllo versione vengono utilizzati per la gestione del codice sorgente e della documentazione a corredo, mai per la gestione dei materiale binario come il codice compilato o altre risorse.

A supporto del progetto DPS è stato utilizzato l'efficace sistema di controllo versione subversion[i].

Documentazione

Il progetto DPS è caratterizzato da una ingente produzione di documentazione. Il codice Java prodotto è stato corredato di commenti sensibili i quali possono essere raccolti in modo automatico e riprodotti nel noto formato Java Doc.

Le riunioni tecniche e le varie fasi di progettazione hanno prodotto numerose considerazioni da riportare in un sistema unificato di documentazione, in particolare a scopo di consultazione nel corso della programmazione. Per questo tipo di informazioni è possibile realizzare un "sito di documentazione". Un sito di documentazione consente di raccogliere il materiale documentale prodotto in qualunque fase del progetto e soggetto a controllo versione per generare automaticamente un sito web navigabile che riporta ogni dettaglio.

Nel corso del progetto è stato utilizzato lo strumento Apache Forrest[ii]. Forrset consente di specificare le informazioni in un conveniente formato XML. Le pagine di documentazione vengono incluse in un progetto di sito, del quale è possibile configurare il tema e la struttura. Ogni pagina XML generata sarà convertita in un formato conveniente alla visualizzazione attraverso un web browser.

Per visualizzare rapidamente le modifiche, Forrest dispone di un server Jetty[iii] integrato, il quale realizza al volo il sito di destinazione interpretando i file XML. Al termine del proget-

---

i Sorgenti, binari e documentazione disponibili presso http://subversion.tigris.org/

ii Sorgenti, binari e documentazione disponibili presso http://forrest.apache.org/

iii Jetty è un server web e servlet container Open Source scritto in Java. Sorgenti, binari e documentazione disponibili presso http://jetty.mortbay.org/index.html

to è possibile compilare l'intero sito in un web archive da eseguire su un servlet container come Jakarta Tomcat[iv] o Jetty stesso.

Forrest costituisce anche un eccellente strumento per il publishing rapido, disponendo di una funzione di generazione di file pdf o postscript dal sito di documentazione.

[iv] Tomcat è un altro server web e servlet container Open Source scritto in Java. Sorgenti, binari e documentazione disponibili presso http://tomcat.apache.org/

Bibliografia

RFC per il protocollo LDAP

[1] Barker, P. and Kille, S. *The COSINE and Internet X.500 Schema*, 1991

[2] Hardcastle-Kille, S. E. *X.500 and Domains*, 1991

[3] Yeong, W. and Howes, T. and Kille, S. *Lightweight Directory Access Protocol*, 1995

[4] Young, A. *Connection-less Lightweight X.500 Directory Access Protocol*, 1995

[5] Howes, T. and Smith, M. *The LDAP Application Program Interface*, 1995

[6] Howes, T. and Smith, M. *An LDAP URL Format*, 1996

[7] Wahl, M. and Howes, T. and Kille, S. *Lightweight Directory Access Protocol (v3)*, 1997

[8] Wahl, M. and Coulbeck, A. and Howes, T. and Kille, S. *Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions*, 1997

[9] Grimstad, A. and Huber, R. and Sataluri, S. and Wahl, M. *Naming Plan for Internet Directory-Enabled Applications*, 1998

[10] Wahl, M. and Howes, T. *Use of Language Codes in LDAP*, 1999

[11] Stokes, E. and Byrne, D. and Blakley, B. and Behera, P. *Access Control Requirements for LDAP*, 2000

[12] Wahl, M. and Alvestrand, H. and Hodges, J. and Morgan, R. *Authentication Methods for LDAP*, 2000

[13] Wahl, M. *MIME Directory Profile for LDAP Schema*, 2000

[14] Zeilenga, K. *LDAP Password Modify Extended Operation*, 2001

[15] Zeilenga, K. *Named Subordinate References in LDAP Directories*, 2002

[16] Zeilenga, K. and Legg, S. *Subentries in the Lightweight Directory Access Protocol*, 2003

Altri documenti

[17] The OpenLDAP Project, *OpenLDAP Software 2.3 Administrator's Guide*, 2005, [Online] http://www.openldap.org

[18] The Mule Project, *Mule User Guide*, 2006, [Online] http://mule.mulesource.org/

[19] Kuhner, J. (editor), *Tapestry Wiki*, 2006, [Online] http://wiki.apache.org/tapestry

[20] The Hivemind Project, *Hivemind Overview / Hivemind Project Informations*, 2006, [Online] http://jakarta.apache.org/hivemind/

Ringraziamenti

Dedico quest'opera e gli sforzi che l'hanno portata a compimento ai miei familiari, senza il cui supporto non avrei potuto sostenere gli studi che mi hanno condotto fino a questo punto.

Desidero inoltre ringraziare tutti i colleghi e amici del Centro di Documentazione, nel cui ambiente è stato sviluppato il progetto trattato in questo documento. Ringrazio inoltre i collaboratori di Datacode srl per il loro ruolo fondamentale nella realizzazione del prodotto finale.

Un ringraziamento particolare va alla mia fidanzata Maria Giovanna, che mi ha accompagnato nei momenti più belli e aiutato, guidato e sorretto nei momenti più difficili.

Licenza del documento

creative commons

COMMONS DEED

Attribuzione - Non commerciale 2.5

Tu sei libero:

- di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera
- di modificare quest'opera

Alle seguenti condizioni:

Devi attribuire la paternità dell'opera nei modi indicati dall'autore o da chi ti ha dato l'opera in licenza.

Non puoi usare quest'opera per fini commerciali.

- Ogni volta che usi o distribuisci quest'opera, devi farlo secondo i termini di questa licenza, che va comunicata con chiarezza.
- In ogni caso, puoi concordare col titolare dei diritti d'autore utilizzi di quest'opera non consentiti da questa licenza.

Le utilizzazioni consentite dalla legge sul diritto d'autore e gli altri diritti non sono in alcun modo limitati da quanto sopra.

http://creativecommons.org/licenses/by-nc/2.5/deed.it

Licenza del software

Progetto Bellerofonte

Il codice sorgente e le configurazioni software prodotte per il Progetto Bellerofonte sono di proprietà del CeDoc Modena e di Datacodo s.r.l. L'autore è autorizzato alla divulgazione di parti del codice dal direttore del Centro di Documentazione. Il codice sorgente citato in questo documento è divulgato in base a tale autorizzazione.

Progetto Newbackup

Il codice sorgente e le configurazioni del progetto Newbackup sono di proprietà del CeDoc Modena e rilasciati sotto licenza BSD. Il codice del prodotto non è incluso né citato in questo documento.