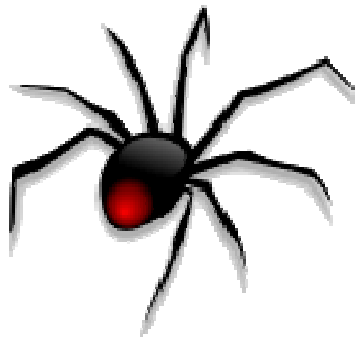


23/07/06 – Rev. C



Spider

Documentazione di Progetto

Gruppo di lavoro

Michelangelo Rinelli <webmaster@tizioandcaio.com>

Luca Leonardo Scordia <luca@skakkinostri.it>

Erjon Shahini <ioniomare@interfree.it>

1. Sommario

1. Sommario	3
2. Introduzione	5
3. Status del documento	5
4. Convenzioni adottate	6
5. Processo di sviluppo.....	6
5.1. Milestones.....	6
5.2. Incarichi	6
5.3. Fonti di complessità.....	7
5.3.1. Multithreading	7
5.3.2. Documentazione	8
5.3.3. Debug procedure concorrenti.....	Errore. Il segnalibro non è definito.
5.4. Ambiente di sviluppo e librerie.....	8
6. Interfaccia utente	9
6.1. Funzionamento generale.....	9
6.2. Il pattern MVC	10
6.3. Finestre di dialogo.....	12
6.3.1. Error pages	12
6.3.2. Add/Edit error pages	12
6.3.3. Edit preferences.....	12
6.3.4. About	13
6.4. Multithreading	14
7. Motore di scansione	15
7.1. Funzionamento generale.....	15
7.2. Classi e collections	15
7.3. Multithreading	16
8. Generazione report.....	17
8.1. Testo semplice.....	17
8.2. Rich Text Format	17
8.3. Microsoft Word	17
8.4. OpenDocument.....	18
9. Diagrammi dei package.....	19
9.1. Package control	19
9.2. Package data.....	20
9.3. Package gui.....	21
9.4. Package report	22
9.5. Package utils	23
10. Testing.....	24
10.1. Il sito web di test	24
10.2. Test run.....	25
10.2.1. SpiderTest	25
10.2.2. www.google.com	26
10.2.3. www.tizioandcaio.com	27
10.2.4. www.skakkinostri.it.....	28
11. Deployment.....	29
12. Formato dei file.....	29
12.1. File di progetto	29
12.2. File di configurazione.....	30
12.2.1. Configurazione Spider.....	30
12.2.2. Configurazione log4j.....	30
13. Possibili sviluppi futuri.....	31
14. Log di SourceGear Vault.....	31

2. Introduzione

Questo documento presenta le scelte architettoniche e progettuali ideate per la realizzazione del software illustrato in analisi. Particolare enfasi viene data alla separazione del software in moduli quanto più possibile indipendenti e alle loro reciproche interazioni. Una sezione apposita discute le problematiche causate dalla necessità di multithreading e agli accorgimenti tecnici necessari. Infine, si presentano le problematiche relative alla generazione dei report nei diversi formati e la sintassi dei file di progetto e di configurazione.

3. Status del documento

Questo documento è il risultato di diverse revisioni. Si riporta di seguito una breve cronologia delle modifiche più importanti.

- ✓ 23/07/2006 – Rev. C
Aggiunta sezione Deployment, completata sezione testing
- ✓ 17/07/2006 – Rev. B
Definite convenzioni e processo di sviluppo
- ✓ 15/07/2006 – Rev. A
Stesura iniziale del documento, individuazione contenuti.

4. Convenzioni adottate

Per consentire una più facile integrazione del lavoro dei tre diversi sviluppatori, sono state decise alcune convenzioni nella scrittura del codice. In particolare:

- Il codice dovrà essere separato nei tre package *data*, *control* e *gui*, per separare logicamente i tre livelli di interazione con il sistema (accesso ai dati, logica di applicazione e interfaccia utente);
- I nomi dei package sono tutti in lettere minuscole;
- I nomi delle classi vengono scritti secondo la convenzione CamelCase con iniziale maiuscola;
- I nomi dei metodi e delle proprietà vengono scritti secondo la convenzione camelCase;
- I nomi delle variabili membro sono scritte secondo la convenzione *_camelCase* preceduta da un underscore;
- Tutte le classi non triviali devono essere dotate di un logger;
- Tutte le informazioni potenzialmente rilevanti ai fini del debug devono essere stampate attraverso il logger specifico di ogni classe;
- I checkin inseriti nel software SCC (Source Code Control) devono mantenere il software in uno stato compilabile.

5. Processo di sviluppo

Il modello di riferimento per lo sviluppo del software è a cascata, poiché è facile suddividere il lavoro da compiere in tre stadi di lavorazione, ognuno dei quali può essere realizzato solo una volta pronto il precedente. Le tre fasi individuate sono l'interfaccia grafica, il motore di scansione e la generazione del report.

Tuttavia alcuni passaggi hanno richiesto un feedback tra realizzazione e progetto, poiché particolari ostacoli in fase implementativa sono stati risolti modificando alcune specifiche di progetto.

5.1. Milestones

Il processo di sviluppo viene suddiviso in attività, ciascuna delle quali è assegnata ad un singolo sviluppatore.

1. Prototipo dell'interfaccia utente	Luca
2. Studio di un algoritmo di scansione	Michelangelo
3. Modellazione architettura UI	Luca
4. Realizzazione algoritmo di scansione	Michelangelo
5. Realizzazione sito web di test	Luca
6. Ottimizzazione performances	Luca
7. Reportistica	Erjon
8. Testing estensivo	Erjon

La suddivisione in milestone permette di verificare rapidamente il progresso dello sviluppo.

5.2. Incarichi

Poiché l'analisi ha rilevato che il progetto da realizzare rientra nella categoria dei progetti medio-piccoli, ed il numero di componenti della squadra è di sole tre persone, non ci sono stati dei ruoli ben definiti (statici); una struttura dinamica è stata senz'altro più adatta. In particolare, i ruoli sono stati ricoperti praticamente da tutti:

Capogruppo

In generale per comune consenso il ruolo di capogruppo è stato assunto da Luca Leonardo Scordia che si è occupato di indirizzare la risoluzione dei problemi principali durante l'implementazione del sistema.

Integratori

Tale ruolo è stato assunto dall'intero team. Essi si impegnano nell'attivazione del sistema e nell'inserimento dei dati con i committenti del software.

Programmatori

Dato che i moduli di cui si compone il software sono tre, lo sviluppo è stato ripartito su tutti e tre i membri del team. Michelangelo Rinelli si è occupato dell'implementazione del motore di scansione, Luca Leonardo Scorcìa si è occupato dell'implementazione dell'interfaccia grafica e della ottimizzazione multithread e Erjon Shahini della reportistica.

Analisti

I requisiti ed il modello sono stati inizialmente analizzati da Michelangelo Rinelli e Luca Leonardo Scorcìa, poi successivamente Erjon Shahini e Luca Leonardo Scorcìa si sono occupati della revisione dell'analisi durante gli incontri con l'ing. R. Totaro, committente del progetto.

Tester

Ogni sviluppatore ha condotto testing informale sul codice scritto di suo pugno, mentre i test più sistematici – con annessa scrittura del sito web di test – sono stati assegnati a Luca Leonardo Scorcìa in qualità di capogruppo e responsabile dell'integrazione dei vari moduli.

Garante della qualità:

Principalmente di questo ruolo si è occupato Erjon Shahini che ha verificato la correttezza dei processi e delle procedure di sviluppo del software.

5.3. Fonti di complessità

Le principali difficoltà che si prevede di incontrare nel corso dello sviluppo riguardano l'integrazione dei diversi componenti del software e la necessità di introdurre elementi di programmazione concorrente (multithread) per migliorare le performances del software.

5.3.1. Multithreading

Poiché il software dovrà interrogare un server web, gran parte del tempo di elaborazione sarà impegnata in attesa di dati dal server. Se ci mantenessimo in una architettura a thread singolo, l'interfaccia utente resterebbe bloccata per tutto il tempo di trasferimento delle pagine e ciò chiaramente è inaccettabile: l'utente sarebbe portato a pensare che il software è inaffidabile. D'altra parte, nemmeno una struttura a due thread (UI e lavoro) è sufficiente! Infatti i server web sono capaci di servire ben più di una sola richiesta alla volta, e non sfruttare questa possibilità significherebbe porre un limite inutile alla velocità di elaborazione del nostro programma. Pertanto si è scelto di implementare una architettura completamente multithread:

- Un thread di interfaccia utente;
- Un thread di controllo di elaborazione;
- Un thread per la comunicazione tra i primi due thread;
- Un certo numero di thread (configurabile dall'utente secondo le capacità del server remoto) per il recupero e l'analisi delle pagine.

L'architettura così ottenuta è estremamente efficiente: tutti questi thread sono sincronizzati in modo da rimanere per gran parte del tempo inattivi (l'operazione più onerosa in termine di tempo è il recupero delle pagine dal server web, ovvero una operazione di I/O). Ciò permette di lasciare la CPU del sistema client inutilizzata per la stragrande maggioranza del tempo. Sul lato server, la possibilità di richiedere più pagine contemporaneamente riduce il tempo di analisi del sito, minimizzando il tempo richiesto per l'esecuzione del test.

La milestone di ottimizzazione, durante la quale i thread sono stati sincronizzati al fine di ridurre al minimo l'uso della CPU client, ha portato un incremento delle performances di esecuzione superiore al 600%!

Sito di test: www.google.com – 5 threads di elaborazione	
Prima della milestone ottimizzazione	120 pagine/minuto
Dopo la milestone ottimizzazione	770 pagine/minuto
Variazione	642%

Naturalmente miglioramenti così elevati si possono ottenere su siti ben forniti di banda e risorse di calcolo, come per l'appunto google.com. Tuttavia è un indice della bontà dei nostri algoritmi.

5.3.2. Documentazione

E' facilmente prevedibile che l'artificiosa complessità introdotta dal generatore di codice dell'interfaccia utente di Eclipse causerà qualche problema durante la documentazione del codice. Per affrontare e superare tali problemi si cercherà di privilegiare una metodologia razionale, basata sull'effettiva utilità della documentazione e più orientata alla descrizione dei concetti di fondo che al commento di dettaglio del codice.

5.3.3. Testing e debugging

Una delle maggiori problematiche nello sviluppo di Spider sarà quella di testare il corretto funzionamento dell'applicazione e di correggere eventuali bug. L'applicazione infatti dovrà essere abbastanza robusta poiché dovrà prevedere non solo il corretto funzionamento in tutti gli scenari corretti possibili, ma dovrà far fronte anche ad eventuali problemi dovuti alle comunicazioni client-server e soprattutto ad errori sintattici nelle web application che dovranno essere sottoposte a visita.

Si prevede quindi che risulterà molto onerosa la fase di testing ed eventuali sessioni di debugging, soprattutto poiché gli scenari possibili di utilizzo sono virtualmente illimitati.

5.4. Ambiente di sviluppo e librerie

Come ambiente di sviluppo si adotta Eclipse IDE 3.1, con target il JDK 1.5 (essenzialmente per il supporto alle collection templatizzate).

Per supportare in modo efficiente il debugging delle classi multithreaded si ritiene importante usare sin dal principio una libreria di logging. La scelta è caduta naturalmente sulle celebri librerie log4j, nella versione 1.2.12, che per la loro elevata flessibilità sono sicuramente le più usate in questo campo.

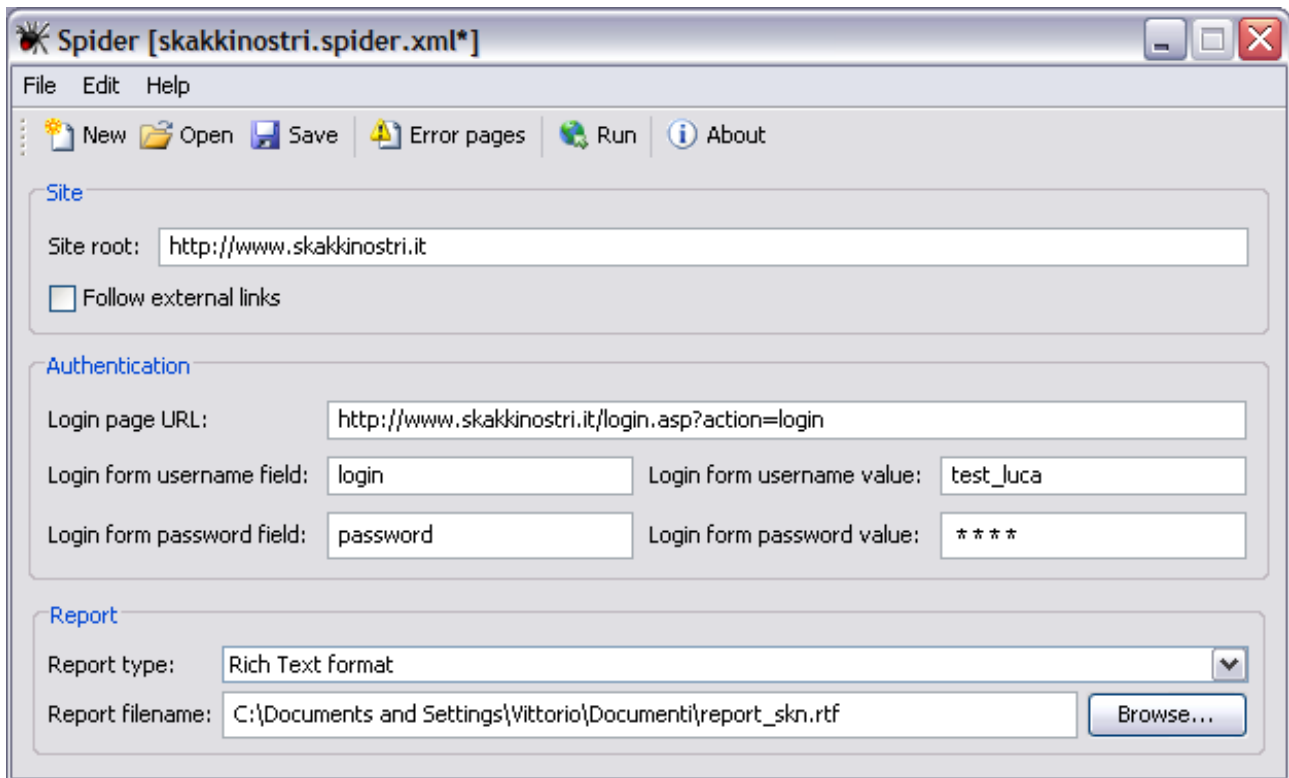
L'interfaccia grafica sarà scritta usando le librerie Swing, mentre non si prevede di utilizzare librerie di terze parti per la generazione dei report nei diversi formati.

Per il testing informale durante lo sviluppo dell'applicazione verrà utilizzato il server web "Microsoft IIS 5.1": tale decisione è basata sull'assenza di vincoli di specifica per tale componente dell'ambiente di sviluppo e sul fatto che tale server web è già installato sulla workstation di uno dei componenti del gruppo.

6. Interfaccia utente

6.1. Funzionamento generale

Il programma si presenta all'utente con una finestra principale (*Main Form*), il cui compito è di fornire all'utente la possibilità di inserire i dati di progetto necessari per eseguire una scansione. Il Main Form inoltre contiene una barra di menu e una toolbar per consentire l'accesso all'utente alle funzionalità di contorno (gestione files di progetto, modifica preferenze).



L'interfaccia utente è progettata per consentire all'utente di avere sotto controllo con una sola occhiata lo stato del programma. Piccoli dettagli, come il riportare il nome del progetto nella barra del titolo, o l'uso di un "*" nella barra del titolo se il documento è stato modificato, possono semplificare di molto la vita dell'utente. Si cerca anche di conformarsi alle linee guida Microsoft per lo sviluppo di interfacce utente intuitive: i pulsanti e le voci di menu che aprono una finestra di dialogo terminano con "...", e a tutti i comandi usati frequentemente sono associate combinazioni di tasti rapidi.

Sono state ideate e realizzate anche alcune comodità per la gestione dei file di documento. In particolare, nel menu File è presente la voce di menu *Open Recent*, che permette all'utente di aprire rapidamente gli ultimi files utilizzati; inoltre il programma chiede una conferma all'utente prima di uscire se sono stati apportate variazioni non salvate al file di progetto.

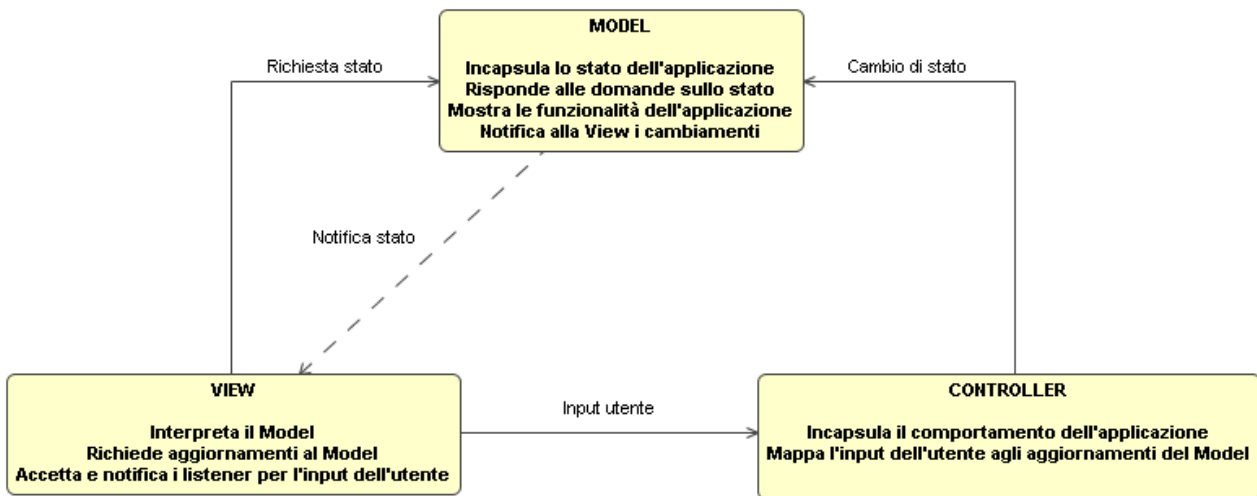
Il programma, come si può vedere, è localizzato in inglese: si è preferito adottare questa lingua piuttosto che tradurre ogni termine tecnico in un italiano approssimativo. D'altra parte esso si rivolge essenzialmente a un utente sviluppatore, che ha buona padronanza dell'inglese tecnico.

6.2. Il pattern MVC

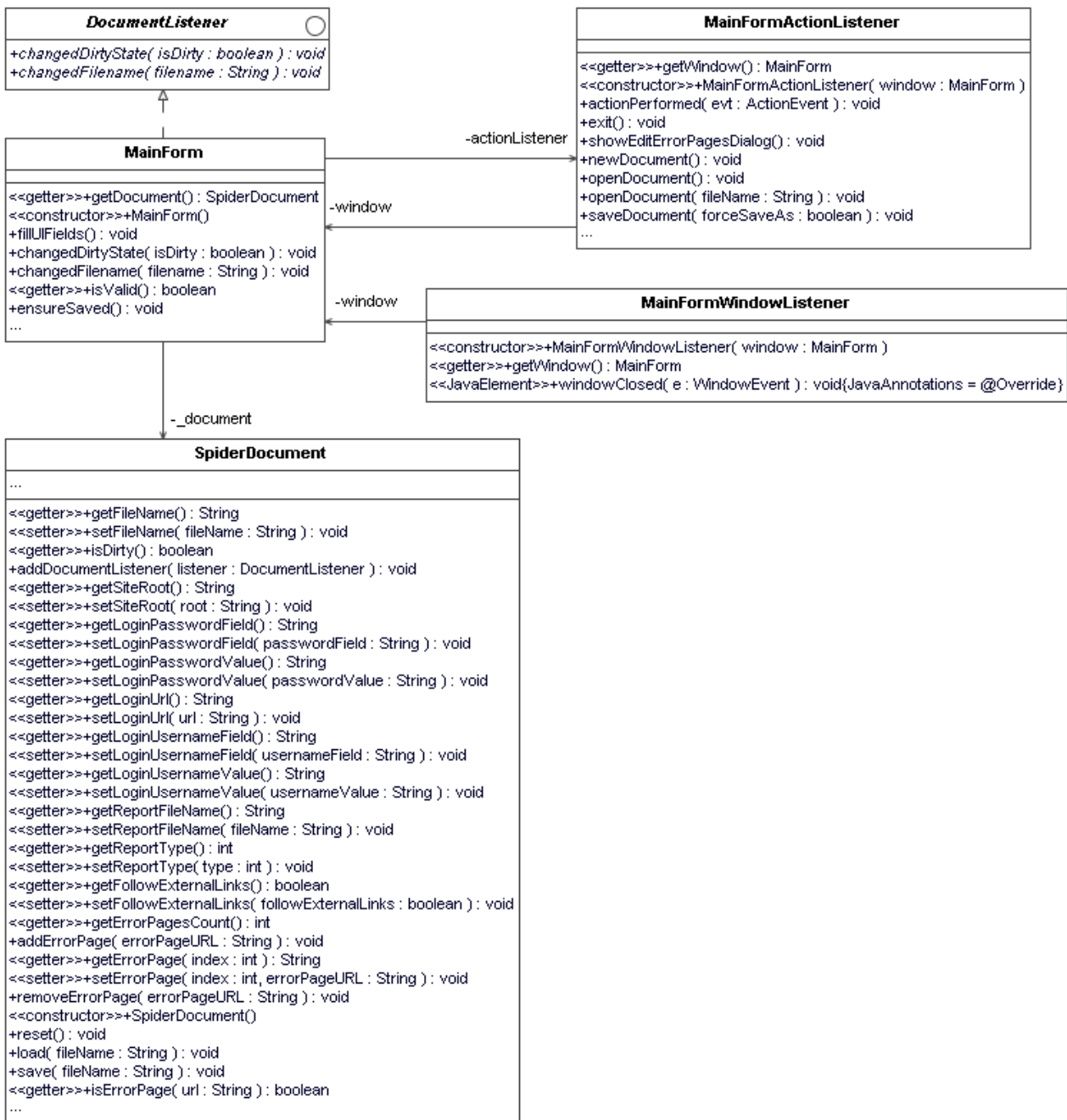
L'applicazione deve separare i componenti software che implementano il modello delle funzionalità di business, dai componenti che implementano la logica di presentazione e di controllo che utilizzano tali funzionalità. Vengono quindi definiti tre tipologie di componenti che soddisfano tali requisiti:

- il Model, che implementa le funzionalità di business
- la View: che implementa la logica di presentazione
- il Controller: che implementa la logica di controllo

La seguente figura rappresenta un diagramma di interazione che evidenzia le responsabilità dei tre componenti.



In Spider, il pattern MVC è stato modellato mediante la gerarchia di classi rappresentata dal diagramma seguente, nel quale il ruolo del Model è assunto dalla classe SpiderDocument, il ruolo del Controller è dato dalla coppia di classi MainFormActionListener e MainFormWindowListener e il ruolo della View è assunto da MainForm. La View riceve le notifiche di variazione di stato poiché implementa l'interfaccia DocumentListener, registrata come event listener presso il Model.

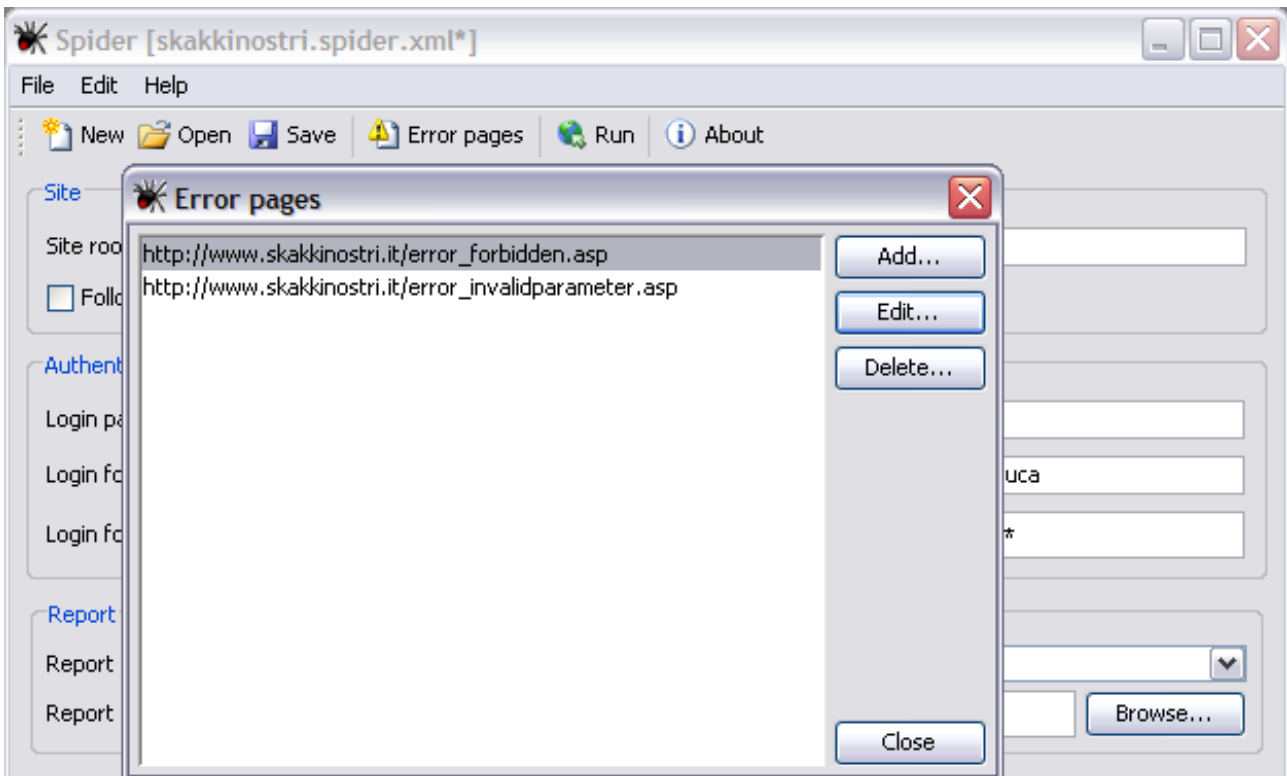


Tramite questa organizzazione è molto facile implementare una serie di comodità per l'utente finale, quale ad esempio la visualizzazione dello status di salvataggio del documento nella barra del titolo dell'applicazione.

6.3. Finestre di dialogo

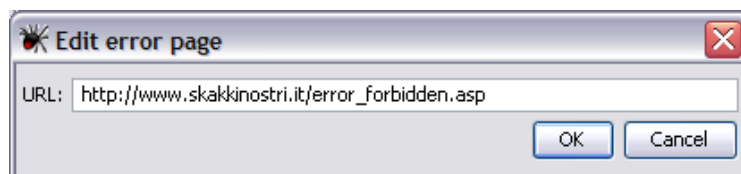
6.3.1. Error pages

La finestra di dialogo Error Pages presenta l'elenco delle pagine che l'applicazione deve trattare come errori durante l'analisi del sito web. I pulsanti Edit e Delete sono attivi solo se è selezionata una voce dalla lista. La finestra è liberamente ridimensionabile a seconda delle necessità dell'utente.



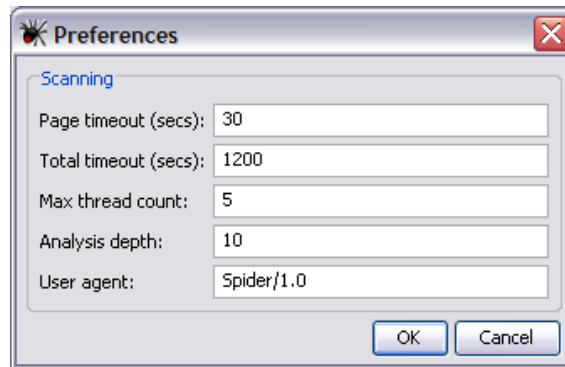
6.3.2. Add/Edit error pages

L'inserimento di nuove voci e la modifica di voci preesistenti nella finestra precedente si compie tramite una apposita finestra di dialogo. Tale finestra di dialogo non permette la chiusura con successo se l'URL inserita non è sintatticamente valida.



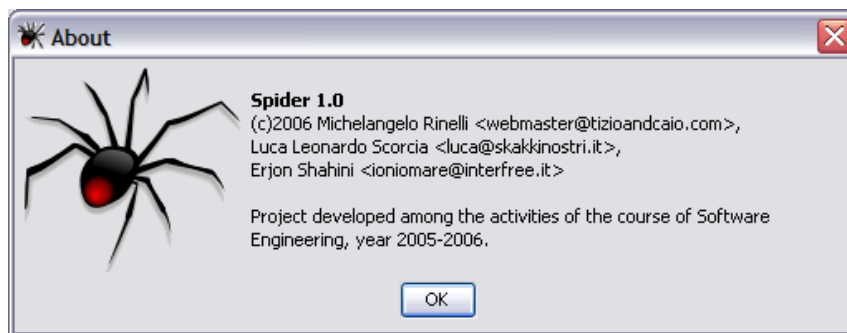
6.3.3. Edit preferences

La finestra di dialogo Preferences contiene i parametri globali di scansione. Tramite apposite caselle di testo è possibile modificare i timeout di recupero pagine, il numero di thread di lavoro da usare, la profondità dell'analisi e l'user agent che utilizzerà lo Spider. La finestra non permette la chiusura se i campi non sono sintatticamente e semanticamente validi.



6.3.4. About

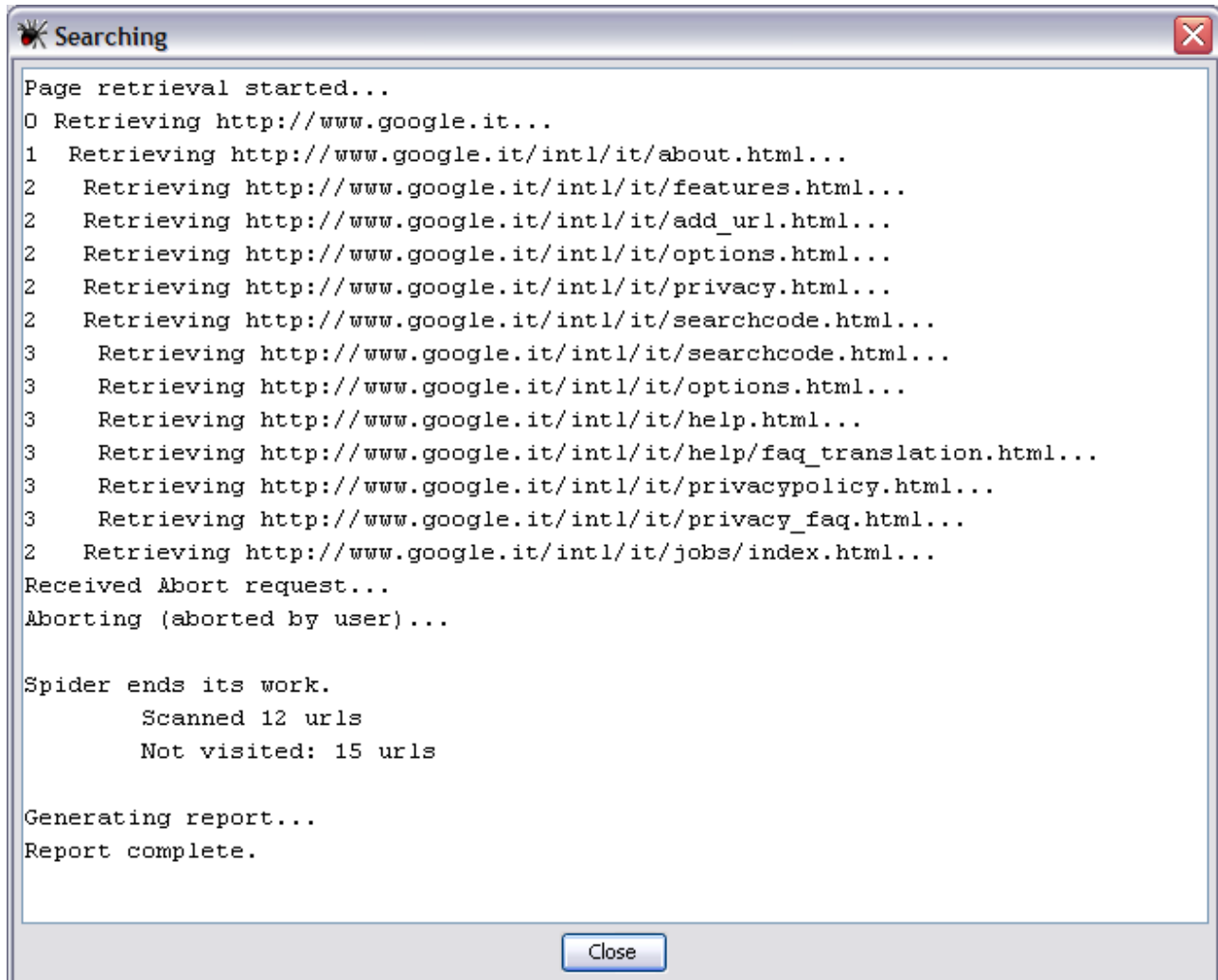
La finestra di About è una semplice presentazione del programma e degli sviluppatori che lo hanno realizzato.



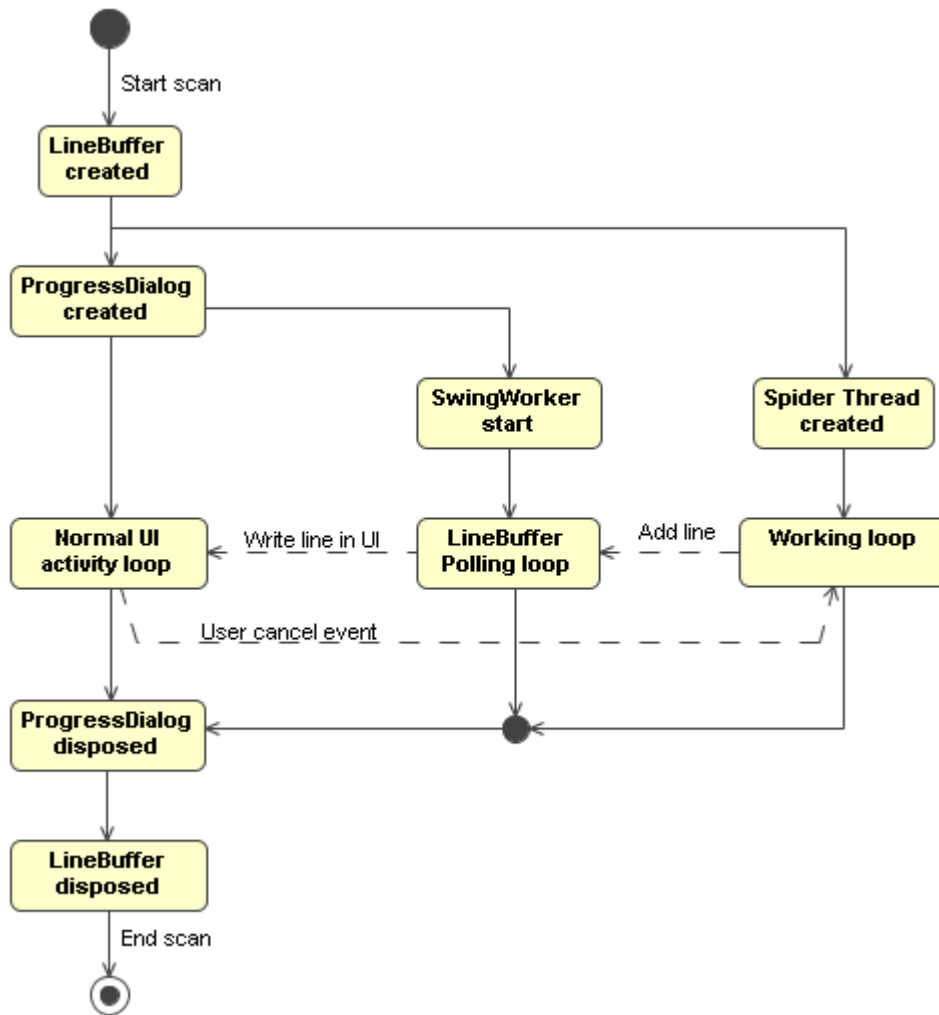
6.4. Multithreading

Le problematiche della gestione multithreaded in una interfaccia utente meritano un piccolo approfondimento. In Swing infatti, solo il thread che ha creato le finestre può manipolarne i controlli. Ciò significa che, senza una opportuna architettura, i thread di lavoro non possono riportare il proprio progresso in finestre di interfaccia utente.

Per superare questo problema, all'interno della documentazione di Java si trova la classe `SwingWorker`, una speciale classe Java che è in grado di manipolare l'interfaccia utente in un thread separato. Tale classe può essere estesa per eseguire il polling di una lista di messaggi e mostrarli in una finestra di dialogo:



Ciò che in realtà sta succedendo mentre questa finestra mostra il progresso dell'operazione può essere ben schematizzato dal diagramma seguente:



Poiché il LineBuffer è una risorsa condivisa da tre thread (in realtà più di tre thread, ma i dettagli del Working loop si possono leggere nella sezione Motore di scansione), di cui due sono di interfaccia utente (Thread principale e SwingWorker), l'accesso ai suoi metodi richiede sincronizzazione.

La possibilità di processare gli eventi di interfaccia in un thread separato rispetto al thread di lavoro consente anche la possibilità per l'utente di terminare la scansione in qualsiasi momento. Per far ciò, viene inviato dal thread di UI al working loop un messaggio di Abort, ed il working loop termina in modo naturale, senza forzare la terminazione (cosa che potrebbe lasciare risorse occupate inutilmente).

7. Motore di scansione

7.1. Funzionamento generale

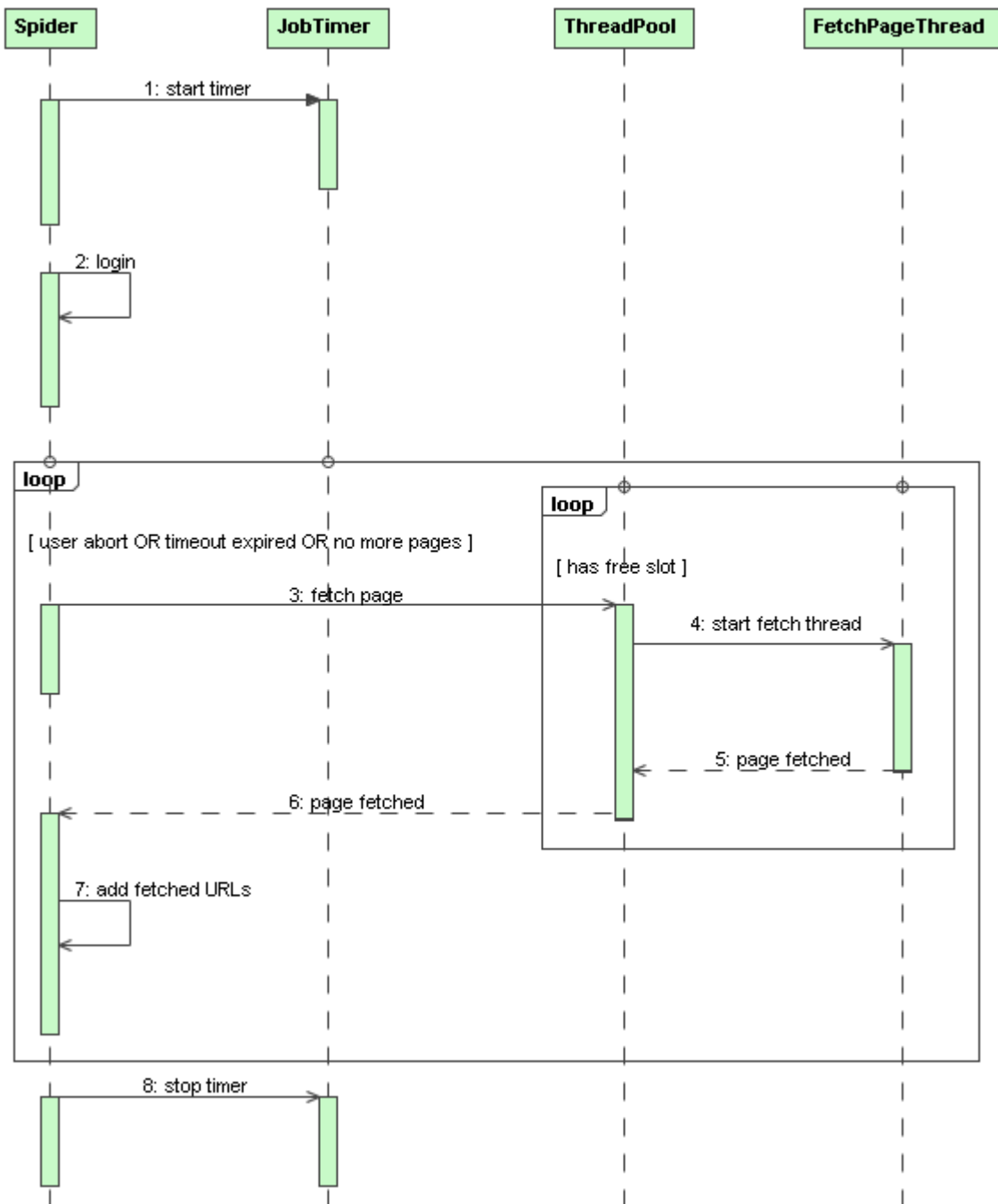
7.2. Classi e collections

7.3. Multithreading

E' possibile descrivere il funzionamento del motore di scansione attraverso un diagramma di sequenza che coinvolge i diversi thread dell'applicazione.

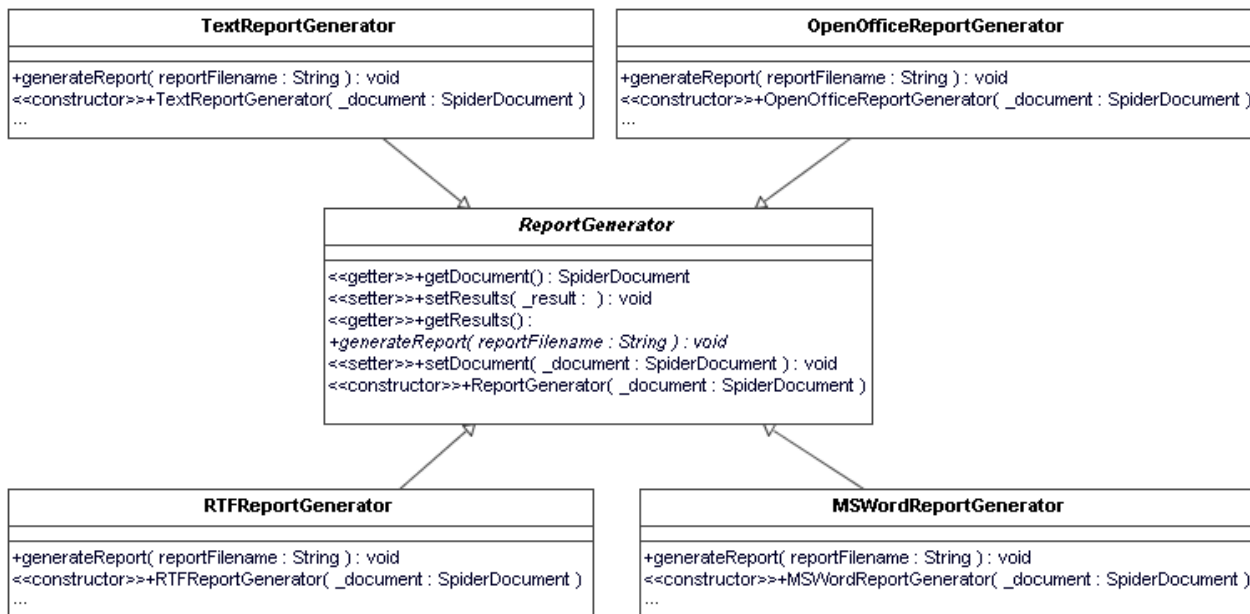
I thread rimangono in attesa durante la maggior parte del tempo di CPU; vengono infatti risvegliati soltanto durante le chiamate dei messaggi qui esposti. Inoltre, l'operazione di recupero pagina all'interno del FetchPageThread è una operazione I/O-bound, quindi ci si può facilmente aspettare che la CPU del sistema client sia decisamente poco impegnata.

In questo diagramma sono omesse le comunicazioni con l'interfaccia utente per facilitare la leggibilità del codice.



8. Generazione report

La creazione dei report al termine della scansione è una attività fondamentale, poiché costituisce il risultato principale che l'utente otterrà. I requisiti prevedono che tali report siano disponibili in quattro formati diversi, ognuno dei quali dovrà essere generato da codice autonomo. Si è pertanto pensato ad una gerarchia di classi come in figura:



I dati che il report deve presentare sono:

- Informazioni sul progetto (nome del progetto, sito esaminato, credenziali adoperate);
- Pagine che hanno causato un errore;
- Pagine che sono state recuperate con successo.

E' possibile vedere esempi dei diversi report generati nella sezione *Test Run*.

8.1. Testo semplice

Il report in file di testo è senz'altro il più semplice. Non vi è alcuna formattazione tranne alcune linee ottenute tramite codici ASCII.

8.2. Rich Text Format

Il formato RTF è un semplice file di testo con pochi, semplici tag di formattazione. In particolare, definiremo il font usato (Times New Roman, Serif), la dimensione, e per i titoli di sezione gli attributi di grassetto e sottolineato.

La scrittura del file RTF non comporta particolari difficoltà, e ci si può basare sull'output di un qualsiasi word processor che supporti il formato RTF. Tuttavia nel seguire questa strada si nota presto come anche editor semplici come WordPad inseriscano tutta una serie di tag non strettamente necessari nel proprio output. Per mantenere compatto il report abbiamo quindi esaminato le specifiche del formato RTF per individuare i tag ridondanti e scrivere solo la formattazione essenziale.

8.3. Microsoft Word

Il formato .doc è in realtà un formato binario e le cui specifiche non sono del tutto pubbliche, il che significa che scrivere documenti in tale formato richiede un impegno notevole.

E' stato però possibile evitare tale sforzo considerando che tutte le versioni di Microsoft Word dalla 6 fino a quella attuale sono in grado di aprire i documenti .doc scritti come testo RTF. Tale formato, denominato Microsoft Word 6/95, è sufficientemente espressivo per i nostri scopi

e ci consente di riutilizzare il codice scritto per il formato RTF. Qualora dovesse essere necessario in futuro ricorrere a formati più sofisticati, si può semplicemente riscrivere tale modulo con il supporto nativo per i formati Microsoft binari più recenti.

8.4. OpenDocument

Il formato OpenDocument è il più complesso tra quelli considerati. Il formato è ampiamente documentato sul web, tuttavia molta della documentazione disponibile è incentrata sulle possibilità avanzate che offre questo formato e non descrive adeguatamente le basi necessarie per la realizzazione di file semplici. A questo problema c'è da aggiungere che il principale software per la modifica di file OpenDocument, OpenOffice, non rispetta pienamente lo standard stesso: infatti, a norma di specifiche, dovrebbe essere possibile creare file elementari direttamente seguendo un formato XML elementare e conveniente; tuttavia l'implementazione di OpenOffice non è in grado di leggerli, in quanto supporta solo i formati di archivio compresso.

E' possibile ricavare un documento OpenDocument minimale salvando un documento vuoto in OpenOffice e procedendo ad eliminare uno alla volta i file contenuti nell'archivio, verificando di volta in volta che OpenOffice sia poi in grado di riaprirlo e visualizzarlo correttamente.

Seguendo questa procedura si ricava che i file .odt sono in realtà package compressi secondo l'algoritmo ZIP, i cui contenuti fondamentali sono due file XML:

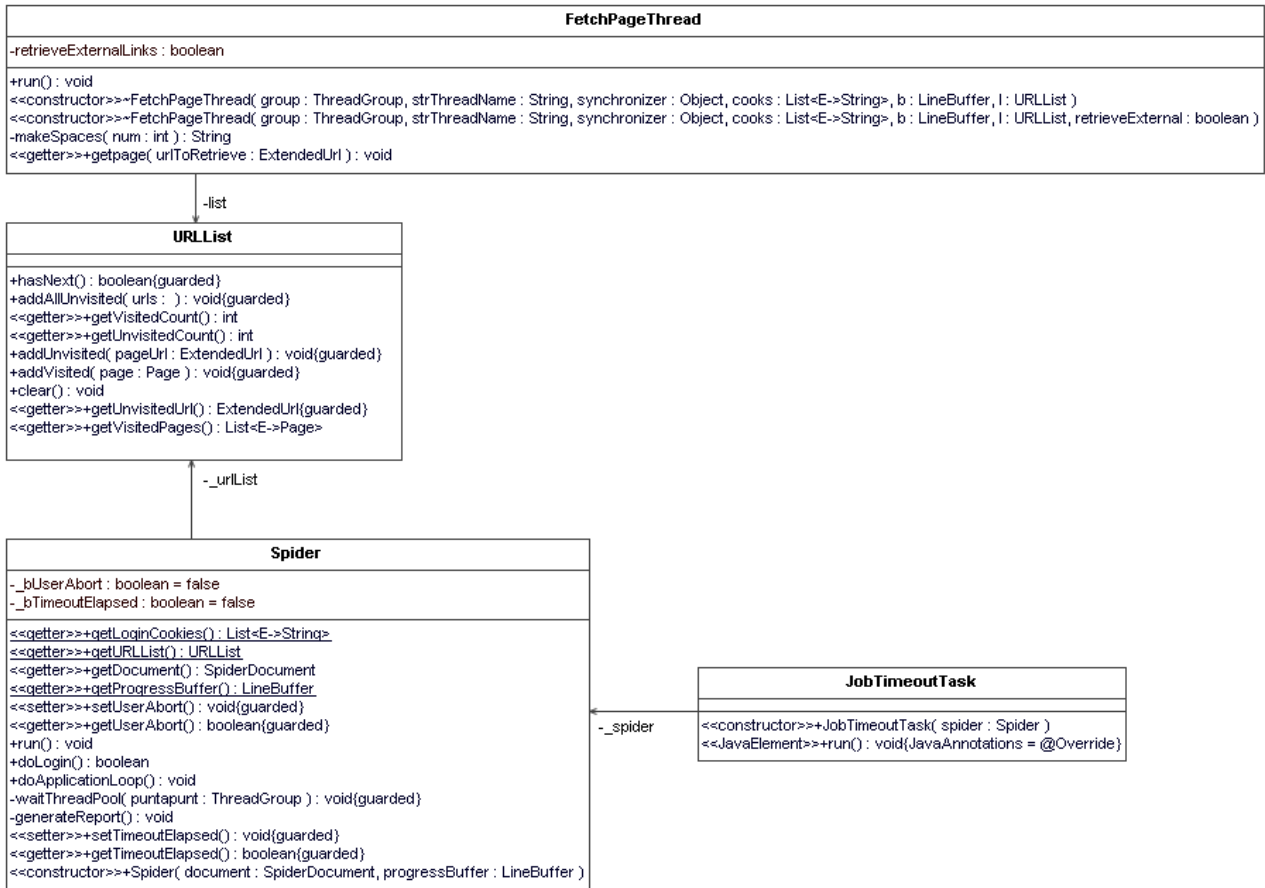
- manifest.xml – contenuto nella sottocartella META-INF – descrive le risorse contenute nel package (ovvero nel nostro semplice caso solo il file content.xml).
- content.xml – il file principale che contiene testo e formattazione – è un documento XML valido che rappresenta l'albero del documento e ne descrive gli stili di formattazione.

Per la compressione degli archivi è possibile utilizzare le librerie java.util.zip, mentre i due file XML dovranno essere generati in memoria e compressi al volo, al fine di evitare la scrittura di inutili file temporanei.

9. Diagrammi dei package

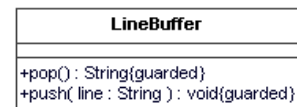
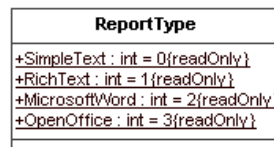
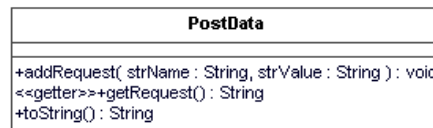
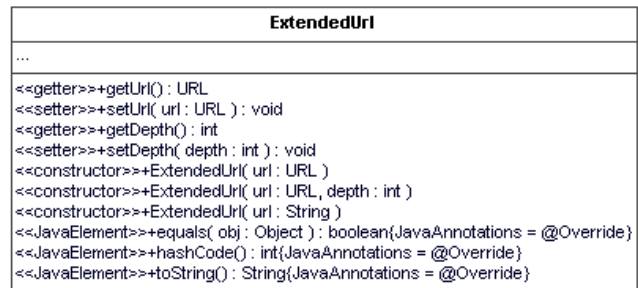
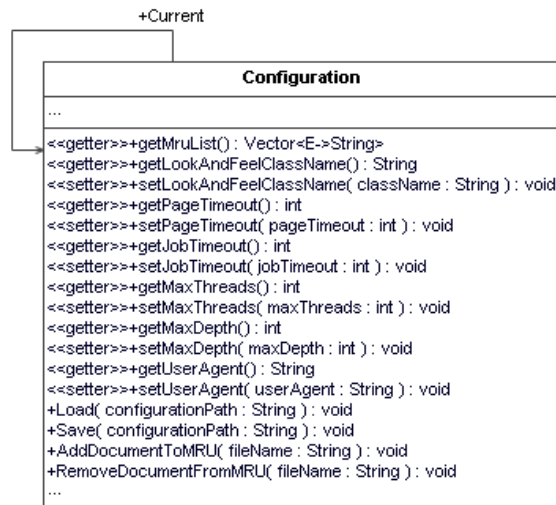
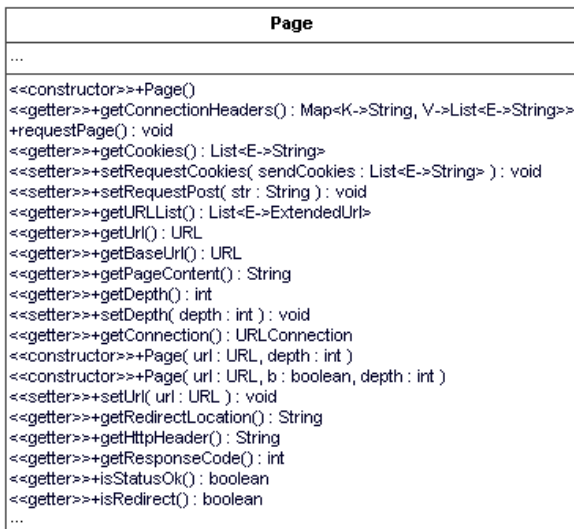
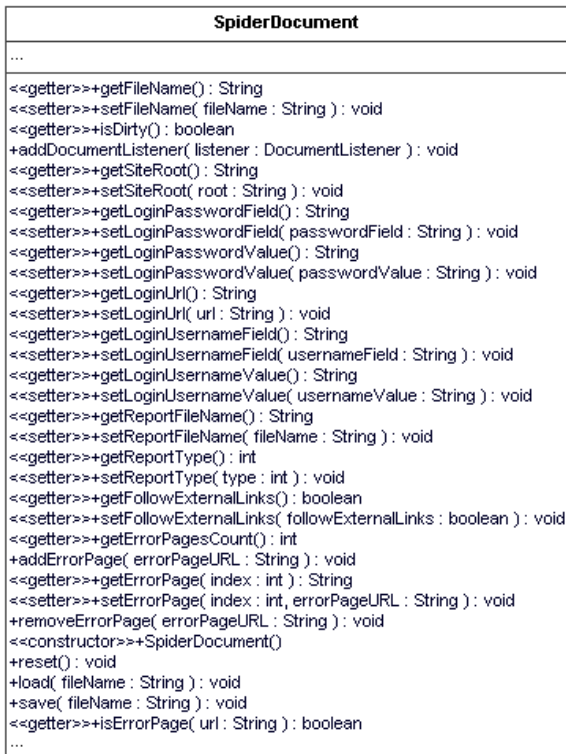
9.1. Package control

Il package control contiene le classi che controllano il flusso di esecuzione del motore di scansione.



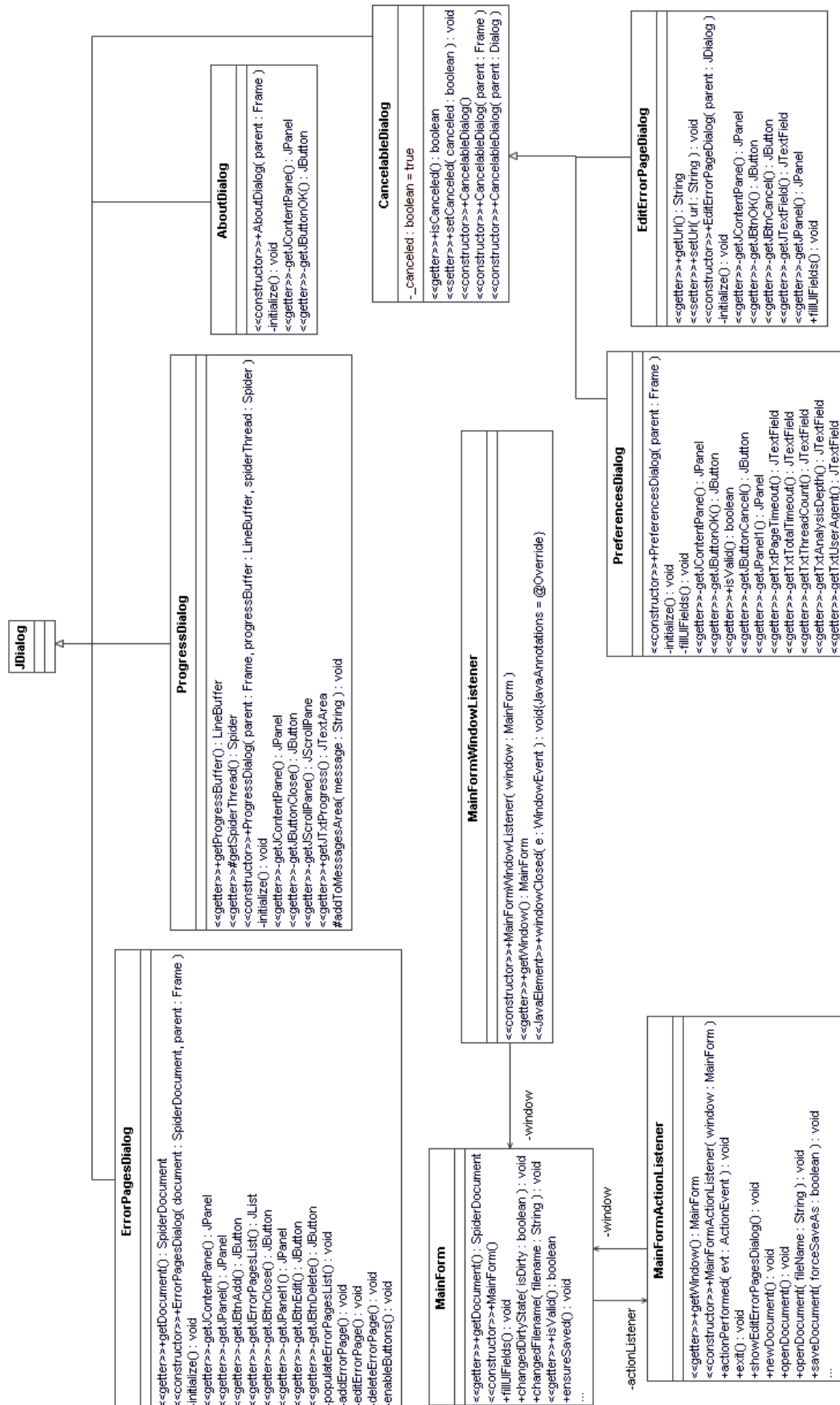
9.2. Package data

Il package data contiene le classi che rappresentano strutture dati.



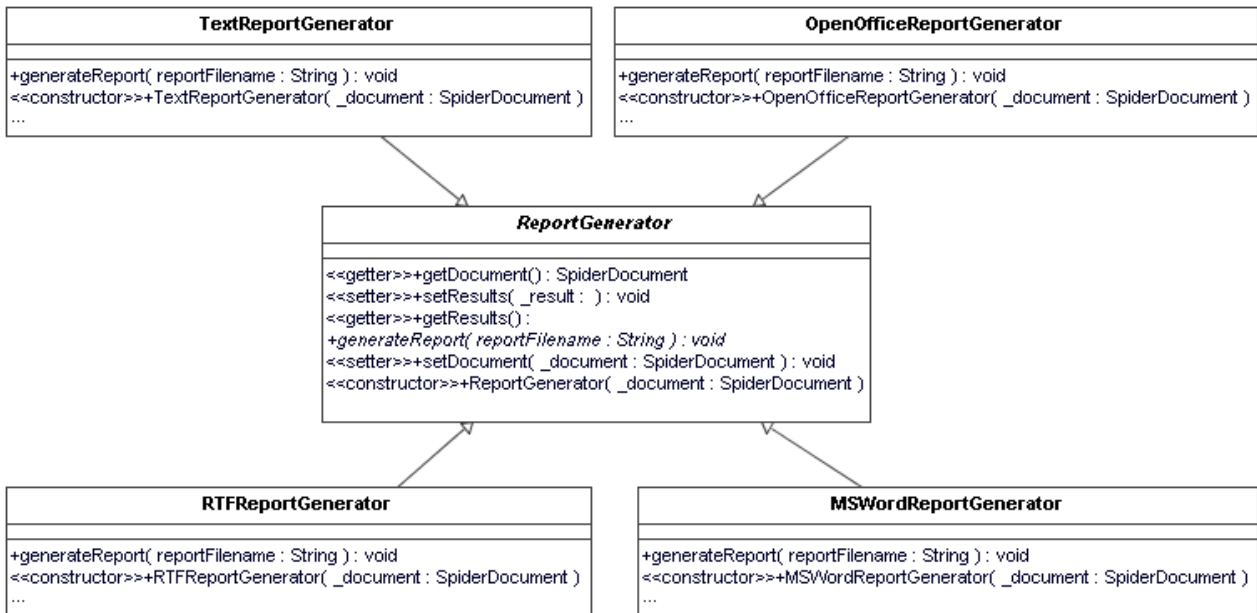
9.3. Package gui

Il package gui comprende le classi che compongono l'interfaccia utente.



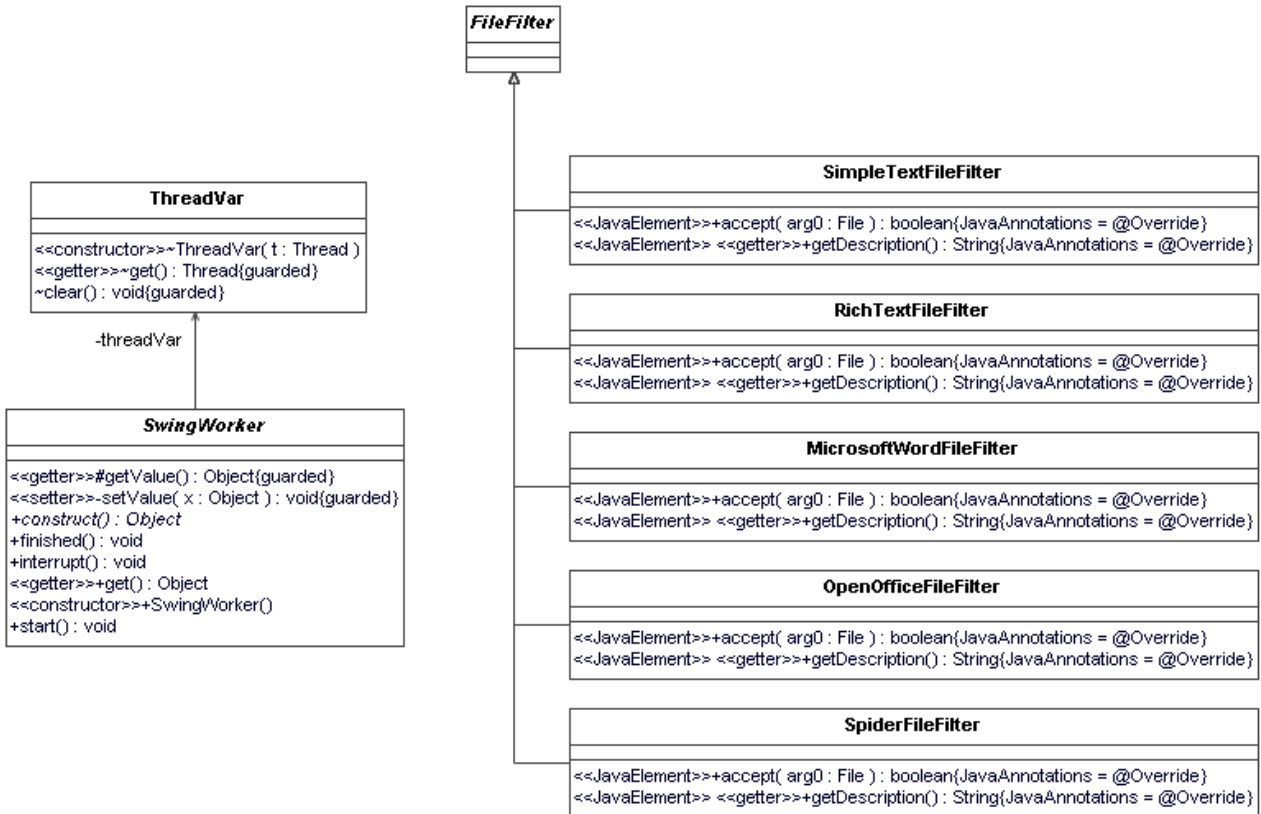
9.4. Package report

Il package report comprende le classi che si occupano di generare i report per l'utente.



9.5. Package utils

Il package utils comprende una serie di classi di comodo, le cui finalità sono generiche e riutilizzabili in diversi progetti.



10. Testing

Durante l'intero ciclo di sviluppo del software è stato condotto testing informale, verificando frequentemente il codice scritto e confrontando a tavolino le soluzioni sviluppate, consultandoci in riunioni oppure via software di Instant Messaging.

Al termine dello sviluppo tuttavia si è ritenuto opportuno condurre una verifica più rigorosa delle funzionalità dell'applicativo, e delle sue reazioni in diverse condizioni ambientali. Innanzitutto si è creato un sito web ad hoc in modo da aderire perfettamente alle specifiche del problema da affrontare, e si è verificato che il comportamento del software fosse quello previsto. Successivamente, si è scelto un piccolo gruppo di siti esistenti nel web sui quali avviare la scansione e verificare che questa andasse a buon fine.

10.1. *Il sito web di test*

Scopo del sito web di test è quello di verificare il corretto funzionamento dei nostri algoritmi. Si tratterà pertanto di un sito di dimensioni molto ridotte, in cui ogni pagina sarà studiata per la verifica di una funzionalità specifica del programma.

In tale ottica, si sono realizzate sette pagine:

- **Default.asp**
Pagina principale del sito, fornisce collegamenti alle diverse pagine di contenuto;
- **Login.asp**
Pagina che contiene i controlli per effettuare il login dell'utente;
- **Everyone.asp**
Pagina accessibile a tutti gli utenti, anche non autenticati;
- **Registered.asp**
Pagina accessibile ai soli utenti autenticati;
- **Admin.asp**
Pagina accessibile ai soli utenti amministratori;
- **Denied_registered.asp**
Pagina di errore che viene mostrata se un utente non autenticato richiede una pagina che necessita di autenticazione;
- **Denied_admin.asp**
Pagina di errore che viene mostrata se un utente non amministratore richiede una pagina che necessita di tali privilegi.

10.2. Test run

10.2.1. SpiderTest

Il sito web di test verrà eseguito con tre credenziali diverse: utente non autenticato, autenticato e amministratore. Per ognuna delle tre esecuzioni dovremo avere risultati diversi.

```
<?xml version="1.0" encoding="UTF-8"?>
<spider>
  <site followExternalLinks="false" root="http://PENTIUM4/SpiderTest/default.asp"/>
  <report filename="C:\Documents and Settings\Vittorio\Desktop\report_spidertest.txt" type="0"/>
  <login url="http://PENTIUM4/SpiderTest/login.asp?action=login">
    <username field="username" value="admin"/>
    <password field="password" value="admin"/>
  </login>
  <errorPages>
    <errorPage url="http://PENTIUM4/SpiderTest/denied_registered.asp"/>
    <errorPage url="http://PENTIUM4/SpiderTest/denied_admin.asp"/>
  </errorPages>
</spider>
```

Run #1: Utente non autenticato

Project info

Project Name: C:\Documents and Settings\Vittorio\Documenti\spidertest.spider.xml
Site Root: http://PENTIUM4/SpiderTest/default.asp
Login Credentials: (none)
Report generated on 2006-07-22 19:24:11

Errors

While retrieving page http://PENTIUM4/SpiderTest/admins.asp: HTTP/1.1 302 Object moved
http://PENTIUM4/SpiderTest/denied_admin.asp
While retrieving page http://PENTIUM4/SpiderTest/registered.asp: HTTP/1.1 302 Object moved
http://PENTIUM4/SpiderTest/denied_registered.asp

Succeeded

HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/login.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/everyone.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/default.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/denied_admin.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/denied_registered.asp

Run #2: Utente autenticato

Project info

Project Name: C:\Documents and Settings\Vittorio\Documenti\spidertest.spider.xml
Site Root: http://PENTIUM4/SpiderTest/default.asp
Login Credentials: normal
Report generated on 2006-07-22 19:39:24

Errors

While retrieving page http://PENTIUM4/SpiderTest/admins.asp: HTTP/1.1 302 Object moved
http://PENTIUM4/SpiderTest/denied_admin.asp

Succeeded

HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/everyone.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/default.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/denied_admin.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/registered.asp

Run #3: Utente autenticato e amministratore

Project info

Project Name: C:\Documents and Settings\Vittorio\Documenti\spidertest.spider.xml
Site Root: http://PENTIUM4/SpiderTest/default.asp
Login Credentials: admin
Report generated on 2006-07-22 19:40:07

Errors

Succeeded

HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/admins.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/everyone.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/default.asp
HTTP/1.1 200 OK http://PENTIUM4/SpiderTest/registered.asp

10.2.2. www.google.com

Questo sito, molto ampio, con un gran numero di pagine ed estremamente rapido in risposta è stato scelto per verificare il corretto funzionamento della sincronizzazione multithread (la sua scansione avvia un gran numero di thread) e della potatura in profondità.

```
<?xml version="1.0" encoding="UTF-8"?>
<spider>
  <site followExternalLinks="false" root="http://www.google.it"/>
  <report filename="C:\Documents and Settings\Vittorio\Desktop\google.odt" type="3"/>
  <login url="">
    <username field="" value=""/>
    <password field="" value=""/>
  </login>
  <errorPages/>
</spider>
```

Project info

Project Name: C:\Documents and Settings\Vittorio\Documenti\google.spider.xml
Site Root: http://www.google.it
Login Credentials: (none)
Report generated on 2006-07-23 10:29:15

Errors

Succeeded

HTTP/1.1 200 OK http://www.google.it/intl/it/help/basics.html
HTTP/1.1 200 OK http://www.google.it/intl/it/searchcode.html
HTTP/1.1 200 OK http://www.google.it/support/bin/topic.py?topic=368
HTTP/1.1 200 OK http://www.google.it/intl/it/help/refinesearch.html
HTTP/1.1 200 OK http://www.google.it/intl/it/terms_of_service.html
HTTP/1.1 200 OK http://www.google.it/intl/it/press/picasa.html
HTTP/1.1 200 OK http://www.google.it/intl/it/terms.html
HTTP/1.1 200 OK http://www.google.it/intl/it/press/pressrelease4.html
HTTP/1.1 200 OK http://www.google.it/support/bin/request.py?contact_type=webmaster&hl=it
HTTP/1.1 200 OK http://www.google.it/support/bin/topic.py?topic=8994
HTTP/1.1 200 OK http://www.google.it/psearch/intl/it/privacy.html
HTTP/1.1 200 OK http://www.google.it/help/faq_accounts.html
HTTP/1.1 200 OK http://www.google.it/help/privacy_fusionph.html
HTTP/1.1 200 OK http://www.google.it/support/bin/topic.py?topic=8996
HTTP/1.1 200 OK http://www.google.it/support/bin/topic.py?topic=8995
HTTP/1.1 200 OK http://www.google.it/intl/it/about.html
HTTP/1.1 200 OK http://www.google.it/intl/it/press/overview.html
HTTP/1.1 200 OK http://www.google.it/intl/it/press/pressrelease7.html
[...]
HTTP/1.1 200 OK http://www.google.it/intl/it/holidaylogos.html
HTTP/1.1 200 OK http://www.google.it/support/bin/answer.py?answer=23852
HTTP/1.1 200 OK http://www.google.it/intl/it/press/pressrelease3.html

10.2.3. www.tizioandcaio.com

Questo sito è stato scelto come test delle procedure di login, poiché usa un articolato sistema di cookie per eseguire l'autenticazione dell'utente.

```
<?xml version="1.0" encoding="UTF-8"?>
<spider>
  <site followExternalLinks="false" root="http://www.tizioandcaio.com"/>
  <report filename="C:\Documents and Settings\Vittorio\Desktop\tizioandcaio.txt" type="0"/>
  <login url="http://www.tizioandcaio.com/profile/login.asp">
    <username field="username" value="micca"/>
    <password field="password" value="schedule"/>
  </login>
  <errorPages />
</spider>
```

Project info

Project Name: C:\Documents and Settings\Vittorio\Documenti\tizioandcaio.spider.xml
Site Root: http://www.tizioandcaio.com
Login Credentials: micca
Report generated on 2006-07-23 10:34:24

Errors

Succeeded

```
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?p=324213
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/posting.php?mode=reply&t=33159
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?p=356745
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/posting.php?mode=reply&t=32898
HTTP/1.1 200 OK http://www.tizioandcaio.com/pre-register.asp
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?t=32811
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/privmsg.php?mode=post&u=13
HTTP/1.1 200 OK http://www.tizioandcaio.com/syndacate.asp
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/posting.php?mode=quote&p=357051
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?p=339555
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/posting.php?mode=quote&p=358593
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?t=32898&view=next
HTTP/1.1 200 OK http://www.tizioandcaio.com/mp3tech_maschera.asp
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/privmsg.php?mode=post&u=23
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?t=32675
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?p=354064
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/profile.php?mode=viewprofile&u=27
[...]
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/profile.php?mode=viewprofile&u=26
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?t=11496&start=0
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?t=33159
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?p=355464
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?t=33044
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?t=31627
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?t=32655
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?p=322637
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?p=356507
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/posting.php?mode=quote&p=357744
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/search.php?search_author=genio
HTTP/1.1 200 OK http://www.tizioandcaio.com/shop/youbuy/category.asp?n=507
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewtopic.php?t=31598
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewforum.php?f=19
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/privmsg.php?mode=post&u=42
HTTP/1.1 200 OK http://www.tizioandcaio.com/staff.asp
HTTP/1.1 200 OK http://www.tizioandcaio.com/musica.asp
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewforum.php?f=51
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/viewforum.php?f=6
HTTP/1.1 200 OK http://www.tizioandcaio.com/shop/youbuy/category.asp?n=36
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/profile.php?mode=resendactivation
HTTP/1.1 200 OK http://www.tizioandcaio.com/rss/threads.xml.asp
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/memberlist.php
HTTP/1.1 200 OK http://www.tizioandcaio.com/public/forum/profile.php?mode=sendpassword
HTTP/1.1 200 OK http://www.tizioandcaio.com/shop/youbuy/category.asp?n=3
HTTP/1.1 200 OK http://www.tizioandcaio.com/staff_login.asp
```

10.2.4. www.skakkinostri.it

Questo sito è stato scelto per testare il riconoscimento delle pagine di errore, vista la grande quantità di pagine non accessibili agli utenti non identificati.

```
<?xml version="1.0" encoding="UTF-8"?>
<spider>
  <site followExternalLinks="false" root="http://www.skakkinostri.it"/>
  <report filename="C:\Documents and Settings\Vittorio\Desktop\skakkinostri.txt" type="0"/>
  <login url="">
    <username field="" value=""/>
    <password field="" value=""/>
  </login>
  <errorPages>
    <errorPage url="http://www.skakkinostri.it/error_forbidden.asp"/>
    <errorPage url="http://www.skakkinostri.it/error_invalidparameter.asp"/>
    <errorPage url="http://www.skakkinostri.it/error_forum_closed.asp"/>
    <errorPage url="http://www.skakkinostri.it/error_forum_non_esistente.asp"/>
    <errorPage url="http://www.skakkinostri.it/error_mailboxfull.asp"/>
    <errorPage url="http://www.skakkinostri.it/error_message_locked.asp"/>
    <errorPage url="http://www.skakkinostri.it/error_multe.asp"/>
    <errorPage url="http://www.skakkinostri.it/error_not_logged_in.asp"/>
    <errorPage url="http://www.skakkinostri.it/error_profile_not_enabled.asp"/>
    <errorPage url="http://www.skakkinostri.it/error_user_blocked.asp"/>
  </errorPages>
</spider>
```

Project info

Project Name: C:\Documents and Settings\Vittorio\Documenti\skakkinostri.spider.xml
Site Root: http://www.skakkinostri.it
Login Credentials: (none)
Report generated on 2006-07-23 11:00:11

Errors

While retrieving page http://www.skakkinostri.it/profile.asp?id_utente=5608: HTTP/1.1 302 Object moved http://www.skakkinostri.it/error_not_logged_in.asp?title=Profilo
While retrieving page http://www.skakkinostri.it/profile.asp?id_utente=2186: HTTP/1.1 302 Object moved http://www.skakkinostri.it/error_not_logged_in.asp?title=Profilo
While retrieving page http://www.skakkinostri.it/profile.asp?id_utente=5287: HTTP/1.1 302 Object moved http://www.skakkinostri.it/error_not_logged_in.asp?title=Profilo
[...]
While retrieving page http://www.skakkinostri.it/profile.asp?id_utente=4008: HTTP/1.1 302 Object moved http://www.skakkinostri.it/error_not_logged_in.asp?title=Profilo
While retrieving page http://www.skakkinostri.it/profile.asp?id_utente=12468: HTTP/1.1 302 Object moved http://www.skakkinostri.it/error_not_logged_in.asp?title=Profilo
While retrieving page http://www.skakkinostri.it/profile.asp?id_utente=16464: HTTP/1.1 302 Object moved http://www.skakkinostri.it/error_not_logged_in.asp?title=Profilo
While retrieving page http://www.skakkinostri.it/profile.asp?id_utente=27875: HTTP/1.1 302 Object moved http://www.skakkinostri.it/error_not_logged_in.asp?title=Profilo

Succeeded

HTTP/1.1 200 OK http://www.skakkinostri.it/article.asp?id_articolo=1028
HTTP/1.1 200 OK http://www.skakkinostri.it/eventi_giorno.asp?data=27/07/2006
HTTP/1.1 200 OK http://www.skakkinostri.it/eventi_mese.asp?data=27%2F06%2F2006
[...]
HTTP/1.1 200 OK http://www.skakkinostri.it/evento.asp?id_evento=290
HTTP/1.1 200 OK http://www.skakkinostri.it/eventi_mese.asp?data=13%2F08%2F2006
HTTP/1.1 200 OK http://www.skakkinostri.it/eventi_giorno.asp?data=10/07/2006
HTTP/1.1 200 OK http://www.skakkinostri.it/eventi_mese.asp?data=08%2F05%2F2006
HTTP/1.1 200 OK
http://www.skakkinostri.it/forum.asp?forumType=Single&id_forum=11&id_msg=282119
HTTP/1.1 200 OK http://www.skakkinostri.it/eventi_settimana.asp?data=05%2F06%2F2006
HTTP/1.1 200 OK http://www.skakkinostri.it/eventi_settimana.asp?data=31%2F07%2F2006
HTTP/1.1 200 OK
http://www.skakkinostri.it/forum.asp?forumType=Single&id_forum=12&id_msg=4360
HTTP/1.1 200 OK http://www.skakkinostri.it/postcards.asp
HTTP/1.1 200 OK http://www.skakkinostri.it/evento.asp?id_evento=317
HTTP/1.1 200 OK http://www.skakkinostri.it/article.asp?id_articolo=1081
HTTP/1.1 200 OK http://www.skakkinostri.it/tip.asp?id_tip=35

11. Deployment

Trattandosi di una applicazione stand-alone, è necessario predisporre una apposita procedura per la sua distribuzione agli utenti. Un progetto Java tradizionale è composto da una serie di file compilati (estensione .class), da posizionare tutti in una stessa cartella, sui quali deve poi essere invocato il runtime specificando la classe che contiene il main. Una migliore soluzione è quella che sfrutta gli archivi Jar, comprimendo ogni progetto in un archivio. Tuttavia, ciò non risolve la necessità di avere più file, uno per ogni libreria.

Poiché una simile struttura di progetto non è affatto intuitiva, si è optato per una soluzione molto più comoda, utilizzando i cosiddetti *Fat Jar*. Un Fat Jar è un archivio compresso che contiene al suo interno tutti i file di programma e le sue dipendenze; è possibile crearli tramite un plugin per Eclipse disponibile gratuitamente (è open source) su SourceForge - <http://fjep.sourceforge.net/> - e chiamato per l'appunto Fat Jar.

Fat Jar si può invocare in Eclipse cliccando con il tasto destro sulla radice della soluzione nella finestra Package Explorer e scegliendo Export > Fat Jar. Al termine del Wizard di creazione, sarà disponibile il file Spider.jar, che può essere richiamato da linea di comando scrivendo al prompt:

```
java -jar Spider.jar
```

Per evitare anche questo ultimo step all'utente finale, si è predisposto un file batch Windows per l'esecuzione di tale comando, così da poter avviare il programma con un semplice doppio clic.

La distribuzione all'utente finale può così avvenire anche mediante un semplice archivio ZIP compresso contenente i file Spider.bat e Spider.jar.

12. Formato dei file

Il formato più usato per rappresentare in modo comprensibile all'utente le informazioni è ormai universalmente XML. Pertanto, si è scelto di creare due formati basati su XML per rappresentare i file letti e scritti da Spider.

12.1. File di progetto

L'estensione predefinita per i file di progetto di Spider è *.spider.xml*. Tale file è strutturato come segue:

```
<?xml version="1.0" encoding="UTF-8"?>
<spider>
  <site followExternalLinks="true" root="http://192.168.2.30"/>
  <report filename="C:\Report.doc" type="2"/>
  <login url="http://192.168.2.30/profile/login.php">
    <username field="username" value="Micca"/>
    <password field="password" value="xxxyyyy"/>
  </login>
  <errorPages>
    <errorPage url="http://www.tizioandcaio.com/errore1.asp"/>
    <errorPage url="http://www.tizioandcaio.com/errore2.php"/>
    <errorPage url="http://www.tizioandcaio.com/errore3.aspx"/>
    <errorPage url="http://www.tizioandcaio.com/errore4.jsp"/>
  </errorPages>
</spider>
```

Tutti i parametri contenuti in questo file sono configurabili attraverso l'interfaccia utente del programma, pertanto non è necessario descriverne il significato nel dettaglio (anche perché è intuitivo comprenderne la funzione).

12.2. File di configurazione

12.2.1. Configurazione Spider

La configurazione di Spider viene salvata all'uscita dal programma in un file chiamato *spider.config.xml*. Tale file, residente nella cartella principale del software, è strutturato come segue:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <lookAndFeel class=""/>
  <scanning>
    <page timeout="30"/>
    <job maxdepth="10" maxthreads="5" timeout="1200"/>
    <useragent value="Spider/1.0"/>
  </scanning>
  <mru>
    <file name="C:\google.spider.xml"/>
    <file name="C:\skakkinostri.spider.xml"/>
    <file name="C:\m_rinelli.spider.xml"/>
    <file name="C:\eyesonff.spider.xml"/>
    <file name="C:\irclab.spider.xml"/>
    <file name="C:\skakkinostri_locale.spider.xml"/>
    <file name="C:\tizioandcaio_mike.spider.xml"/>
    <file name="C:\tizioandcaio.spider.xml"/>
  </mru>
</configuration>
```

Il documento è suddiviso in tre sezioni: delle tre, solo la sezione Scanning è modificabile dall'utente tramite il menu Edit > Preferences, in quanto contiene le impostazioni del motore di scansione.

La sezione LookAndFeel invece, se presente, specifica lo stile Swing dell'applicazione. Si è ritenuto opportuno inserire la possibilità di modificare lo stile dell'interfaccia grafica poiché, secondo la nostra esperienza, spesso sotto Linux lo stile di default (Motif/CDE) è poco user-friendly. Tuttavia, essendo un comando non necessario all'utente comune si è preferito non inserirlo nell'interfaccia grafica.

I valori ammissibili per questo parametro sono:

- Un nome di look and feel Java (Esempio: Motif/CDE)
- Una classe di look and feel Java (Esempio: com.sun.java.swing.plaf.motif.MotifLookAndFeel)

Infine, la sezione MRU (Most Recently Used) memorizza i file di progetto usati di recente, così che l'utente possa ritrovarli nel menu File -> Open Recent.

12.2.2. Configurazione log4j

Log4j è la libreria di logging utilizzata per la diagnostica in Spider. Le impostazioni di logging sono memorizzate nel file *log4j.properties*, un documento di testo semplice. Una configurazione valida per il debug può essere la seguente:

```
# root Logger
log4j.rootCategory=ALL, stdout, logfile

# Console output
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d %5p %t %m (%F:%L)%n

# Log file output
log4j.appender.logfile=org.apache.log4j.FileAppender
```

```
log4j.appender.logfile.File=spider.log
log4j.appender.logfile.append=false
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
log4j.appender.logfile.layout.ConversionPattern=%d %5p %t %m (%F:%L)%n
```

Questa configurazione logga tutti i messaggi di debug sia sulla console (così che possano essere letti nella finestra Console di Eclipse), sia su un file di testo chiamato *spider.log*, facilmente spedibile via posta elettronica allo sviluppatore per una diagnosi più accurata di eventuali problemi.

Il conversion pattern scelto logga le seguenti informazioni:

- `%d` Data e ora attuale, con precisione al millisecondo
- `%5p` Tipo di messaggio
- `%t` Nome del thread
- `%m` Messaggio di debug
- `%F` File sorgente
- `%L` Numero di linea sorgente

Esempio:

```
2006-07-19 11:33:00,859 DEBUG main Creating empty document
(SpiderDocument.java:141)
```

13. Possibili sviluppi futuri

Rimanendo nello stesso ambito applicativo, potrebbe essere utile implementare funzionalità non previste dal committente che possono però rendere più utile il nostro prodotto. Ricordando infatti lo scopo primario di Spider, che è di verificare l'accessibilità delle pagine di un sito web date certe credenziali, potrebbe essere utile confrontare non solo l'indirizzo delle pagine con una serie di pagine di errore specificate, ma anche ricercare all'interno delle pagine determinati termini, inseriti dall'utente, che possono indicare un errore.

Inoltre, per automatizzare il processo di verifica della corrispondenza tra le previsioni del tool di modellazione FSM e i risultati ottenuti dallo Spider potrebbe essere utile pensare a una esportazione dei report in formato XML o comunque *machine-readable*, così che i due software possano essere usati in cascata e produrre in automatico un report di differenze e corrispondenze.

Un'altra possibile estensione potrebbe essere il supporto all'autenticazione http – sebbene meno usata, è tuttavia una possibilità che il nostro software dovrebbe contemplare prima di definirsi completo. In tal caso potrebbe essere necessario valutare le problematiche dovute all'introduzione dell'autenticazione mediante SSL.

14. Log di SourceGear Vault

In questa sezione riportiamo un estratto del log del sistema di controllo codice sorgente con tutti i cambiamenti più significativi apportati all'applicazione. In particolare, sono stati documentati i bug rilevati e si può tracciare il lavoro di ogni singolo sviluppatore. Tuttavia a questo proposito bisogna sottolineare che Erjon Shahini non ha potuto accedere al sistema SCC poiché sprovvisto di connessione internet a larga banda; pertanto i suoi contributi (appartenenti all'area di generazione report) risultano a nome di Luca Leonardo Scorcìa, che ha inserito i file nel sistema per conto di Erjon. Inoltre, anche il codice sviluppato durante gli incontri in Politecnico è stato inserito nell'SCC da Luca poiché non è disponibile alcuna connessione Internet per gli studenti.

In questo documento non compaiono le voci corrispondenti alla creazione dei file, essenzialmente per motivi di spazio.

Query: Recursive changes to `$/src` and all its contents, made by anyone, at any time, including files/folders with any name, sorted by date descending.

name	user	date	version	Action
src/report/MSWordReportGenerator.java <i>Report: added current date-time</i>	luca	22/07/2006 18.10	4	Checked In
src/report/OpenOfficeReportGenerator.java <i>Report: added current date-time</i>	luca	22/07/2006 18.10	4	Checked In
src/report/RTFReportGenerator.java <i>Report: added current date-time</i>	luca	22/07/2006 18.10	4	Checked In
src/report/TextReportGenerator.java <i>Report: added current date-time</i>	luca	22/07/2006 18.10	4	Checked In
src/control/Spider.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	12	Checked In
src/data/Page.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	10	Checked In
src/data/SpiderDocument.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	4	Checked In
src/control/URLList.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	6	Checked In
src/control/FetchPageThread.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	7	Checked In
src/report/MSWordReportGenerator.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	3	Checked In
src/report/OpenOfficeReportGenerator.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	3	Checked In
src/report/ReportGenerator.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	3	Checked In
src/report/RTFReportGenerator.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	3	Checked In
src/report/TextReportGenerator.java <i>Scanning Engine refactored and fully working</i>	luca	22/07/2006 17.36	3	Checked In
src/data/Page.java <i>Page timeout, User Agent, No follow redirects</i>	luca	22/07/2006 9.49	9	Checked In
src/control/Spider.java <i>Job timeout support</i>	luca	19/07/2006 19.40	11	Checked In
src/control/Spider.java <i>Support for depth pruning</i>	luca	19/07/2006 19.22	10	Checked In
src/data/Page.java <i>Support for depth pruning</i>	luca	19/07/2006 19.22	8	Checked In
src/control/URLList.java <i>Support for depth pruning</i>	luca	19/07/2006 19.22	5	Checked In
src/control/FetchPageThread.java <i>Support for depth pruning</i>	luca	19/07/2006 19.22	6	Checked In
src/gui/MainFormWindowListener.java <i>Warning unsaved changes, Bugfix dirty on doc open</i>	luca	15/07/2006 10.41	2	Checked In
src/gui/MainFormActionListener.java <i>Warning unsaved changes, Bugfix dirty on doc open</i>	luca	15/07/2006 10.41	4	Checked In
src/gui/MainForm.java <i>Warning unsaved changes, Bugfix dirty on doc open</i>	luca	15/07/2006 10.41	7	Checked In
src/report/OpenOfficeReportGenerator.java <i>Report generator</i>	luca	14/07/2006 23.58	2	Checked In
src/report/TextReportGenerator.java <i>Report generator</i>	luca	14/07/2006 21.55	2	Checked In
src/report/RTFReportGenerator.java	luca	14/07/2006 21.55	2	Checked In

<i>Report generator</i>				
src/report/ReportGenerator.java	luca	14/07/2006 21.55	2	Checked In
<i>Report generator</i>				
src/report/MSWordReportGenerator.java	luca	14/07/2006 21.55	2	Checked In
<i>Report generator</i>				
src/gui/MainForm.java	luca	14/07/2006 21.55	6	Checked In
<i>Report generator</i>				
src/data/PageResult.java	luca	14/07/2006 21.55	2	Checked In
<i>Report generator</i>				
src/data/Page.java	luca	14/07/2006 20.37	7	Checked In
<i>Risolto bug calcolo URL base quando si parsano URL con path <> ""</i>				
src/control/Spider.java	luca	14/07/2006 20.08	9	Checked In
<i>Report generator</i>				
src/control/URLList.java	tixxio	04/07/2006 17.15	4	Checked In
<i>Double fetched page bug fixed</i>				
src/control/FetchPageThread.java	tixxio	04/07/2006 17.15	5	Checked In
<i>Double fetched page bug fixed</i>				
src/data/Page.java	tixxio	04/07/2006 17.00	6	Checked In
<i>Removing duplicate URLs</i>				
src/data/HTTPResponse.java	tixxio	04/07/2006 16.33	2	Checked In
<i>First embrional version of report generator</i>				
src/control/URLList.java	tixxio	04/07/2006 16.33	3	Checked In
<i>First embrional version of report generator</i>				
src/control/Spider.java	tixxio	04/07/2006 16.33	8	Checked In
<i>First embrional version of report generator</i>				
src/gui/dialogs/ProgressDialog.java	luca	29/06/2006 22.48	8	Checked In
<i>New thread loop abort process</i>				
src/control/Spider.java	luca	29/06/2006 22.48	7	Checked In
<i>New thread loop abort process</i>				
src/data/SpiderDocument.java	luca	29/06/2006 21.57	3	Checked In
<i>Correctly set default document values</i>				
src/data/LineBuffer.java	luca	29/06/2006 15.05	4	Checked In
<i>Migliore meccanismo threading</i>				
src/control/Spider.java	luca	29/06/2006 14.57	6	Checked In
<i>Synchronizing experiments</i>				
src/control/FetchPageThread.java	luca	29/06/2006 14.57	4	Checked In
<i>Synchronizing experiments</i>				
src/data/LineBuffer.java	luca	29/06/2006 0.05	3	Checked In
<i>Multiple locks support</i>				
src/data/Page.java	luca	28/06/2006 23.53	5	Checked In
<i>More flexible parsing algorithm; support for base URL referring</i>				
src/gui/dialogs/ProgressDialog.java	luca	28/06/2006 23.28	7	Checked In
<i>Migliore meccanismo di sincronia</i>				
src/data/LineBuffer.java	luca	28/06/2006 23.28	2	Checked In
<i>Migliore meccanismo di sincronia</i>				
src/data/Page.java	luca	28/06/2006 23.03	4	Checked In
<i>Refactored parsing algorithm</i>				
src/gui/dialogs/ProgressDialog.java	luca	28/06/2006 22.01	6	Checked In
<i>Resized progress dialog</i>				
src/control/FetchPageThread.java	luca	28/06/2006 20.36	3	Checked In
<i>Retreive -> Retrieve</i>				
src/gui/MainForm.java	luca	28/06/2006 20.25	5	Checked In
<i>Fixed timing issues.</i>				
src/gui/dialogs/ProgressDialog.java	luca	28/06/2006 20.25	5	Checked In

<i>Fixed timing issues.</i>				
src/data/Page.java	luca	28/06/2006 20.25	3	Checked In
<i>Fixed timing issues.</i>				
src/data/Configuration.java	luca	28/06/2006 20.25	3	Checked In
<i>Fixed timing issues.</i>				
src/control/URLList.java	luca	28/06/2006 20.25	2	Checked In
<i>Fixed timing issues.</i>				
src/control/Spider.java	luca	28/06/2006 20.25	5	Checked In
<i>Fixed timing issues.</i>				
src/control/FetchPageThread.java	luca	28/06/2006 20.25	2	Checked In
<i>Fixed timing issues.</i>				
src/utils/SwingWorker.java	luca	28/06/2006 12.05	2	Checked In
<i>Added thread names</i>				
src/gui/MainFormActionListener.java	luca	28/06/2006 12.05	3	Checked In
<i>Added thread names</i>				
src/control/Spider.java	luca	28/06/2006 12.05	4	Checked In
<i>Added thread names</i>				
src/log4j.properties	luca	28/06/2006 11.57	2	Checked In
<i>Aggiunto il numero di thread alla stringa di log</i>				
src/gui/MainForm.java	luca	28/06/2006 11.49	4	Checked In
<i>Bugfix in MRU list</i>				
src/data/Configuration.java	luca	28/06/2006 11.49	2	Checked In
<i>Bugfix in MRU list</i>				
src/gui/dialogs/ProgressDialog.java	luca	28/06/2006 11.39	4	Checked In
<i>Graceful working loop end</i>				
src/data/PostData.java	luca	28/06/2006 11.39	2	Checked In
<i>Graceful working loop end</i>				
src/control/Spider.java	luca	28/06/2006 11.39	3	Checked In
<i>Graceful working loop end</i>				
src/gui/dialogs/ProgressDialog.java	luca	27/06/2006 22.59	3	Checked In
<i>User Abort support (Pt. 2)</i>				
src/gui/MainFormActionListener.java	luca	27/06/2006 22.41	2	Checked In
<i>User Abort support (Pt. 1)</i>				
src/gui/dialogs/ProgressDialog.java	luca	27/06/2006 22.41	2	Checked In
<i>User Abort support (Pt. 1)</i>				
src/gui/MainForm.java	luca	27/06/2006 22.32	3	Checked In
<i>Aggiunto flag Follow External Links</i>				
src/data/SpiderDocument.java	luca	27/06/2006 22.32	2	Checked In
<i>Aggiunto flag Follow External Links</i>				
src/data/Page.java	luca	27/06/2006 21.37	2	Checked In
<i>Removed test code, completed search loop</i>				
src/control/Spider.java	luca	27/06/2006 21.37	2	Checked In
<i>Removed test code, completed search loop</i>				
src/gui/MainForm.java	admin	27/06/2006 9.02	2	Checked In
<i>Validazione campi</i>				
src/gui/dialogs/PreferencesDialog.java	admin	27/06/2006 9.02	2	Checked In
<i>Validazione campi</i>				
src/gui/dialogs/ErrorPagesDialog.java	admin	27/06/2006 9.02	2	Checked In
<i>Validazione campi</i>				
src/gui/dialogs/EditErrorPageDialog.java	admin	27/06/2006 9.02	2	Checked In
<i>Validazione campi</i>				