Detailed Design Document

Batch Id – CS85/2006/022

Batch Members

Manjunath T
Niranjan N
Pavan Kumar K

Project Guide

Mr. Suresh S Jamadagni

1. Introduction

1.1 Purpose

This document has been prepared according to IEEE standard 1016-1998.

The purpose of this document is to describe the design and architecture of the ***S.M.A.R.T. Downloader*.**

This design document is prepared to establish a communication between the acquirers (Department of Computer Science, PESIT), users, and the software development team.

1.2 Scope

The ***S.M.A.R.T Downloader*** project seeks to develop a downloading tool that seamlessly integrates with most popular browsers, provides comprehensive error recovery and resume capability to restart broken or interrupted downloads, protects against viruses and other attacks by integrating with anti-virus software.

1.3 Targeted Audience

This Design document is intended to act as a technical reference tool for **developers** involved in the development of ***S.M.A.R.T. Downloader.***

2. System Overview

The objective of the project is to design and implement a downloading tool that is compliant with multiple browsers and provides its users fine-grained control over downloading.

The ***S.M.A.R.T Downloader*** provides automatic file segmentation and multipart download in parallel for faster download and efficient channel utilization.
Dynamic file segmentation is provided for user in case he does not have sufficient storage space in local system . Other additional facility is redirecting the file download to specific network location this makes our product unique.

The user interface design for ***S.M.A.R.T Downloader*** would be such that it is visually appealing and provides least surprise to users . Advanced facilities like comprehensive error recovery and resume capability to restart broken or interrupted downloads due to lost connections, network problems, computer shutdowns, or unexpected power outages are also provided.

3. Design Considerations

3.1 Assumptions and Dependencies

S.M.A.R.T. Downloader assumes nothing about its end users. The product provides adequate interface to match the capabilities of both novice and power users.

S.M.A.R.T Downloader can be used both as a stand-alone product or as a plug-in to a browser.
In the latter case, it requires adequate information about the file that is to be downloaded, from the browser.

3.2 General Constraints

S.M.A.R.T Downloader uses multiple channels in parallel to achieve hi-speed download. The total number of such parallel connections should be limited to a particular value (either user specified or a system calculated value) as some servers do not accept multiple connections from the same client.

The product should be able to seamlessly integrate it with most of the popular web browsers.

The product should match the capabilities of a novice user with advanced features for power users to facilitate faster interactions with the system.

S.M.A.R.T Downloader uses certain windows specific functions such as Win32 APIs for its operation. This makes it difficult to be easily run on operating systems other than windows.

The product should provide least surprise to users through an easy-to-use interface and comprehensive error recovery.

3.3 Development Methods

S.M.A.R.T Downloader uses unified approach (UA) with object oriented design methods and uses UML (Unified Modeling Language) to represent design specifications and to document analysis and the design phases.

S.M.A.R.T Downloader uses the .Net framework for development. The product will be developed using C#.net as the programming language on the Windows Operating System.

3.4 Goals and Guidelines

This section describes the goals, guidelines, principles, or priorities which dominate or embody the design of ***S.M.A.R.T Downloader***.

1. Emphasis on Speed to facilitate faster download-times.
2. Emphasis on efficient bandwidth utilization.
3. Visually appealing and easy-to-use interface with comprehensive error recovery.
4. Provide least surprise to users
5. Provide features that target both power users and novice users with sensible distinction between the two classes of users.
6. Seamlessly integrate with popular browsers.

4. System Architecture

This section provides a high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components.

The system has been partitioned into the following modules to facilitate faster development.

- HTTP Downloader
  Performs downloading tasks using HTTP. Additionally supports multi-part download and mirror management.
- FTP Downloader
  Performs downloading tasks using FTP.
- XML and Webservices
  Performs validation of xml and WSDL documents. Parses the WSDL document and XML Configuration files.

Manages stub creation and parameter passing for webservices.

- Uploader
  Provides file uploading services.
- Scheduler and Timer
  Schedules jobs in the pipeline and manages scheduled downloads.
- Plugin Support
  Facilitates integration with web browsers.
- Thread Control
  Manages downloader threads.
- Regular Expressions
  Filters files against a pattern for selective download.
- Security
  Provides encryption and decryption services.
- Services
  Manages background daemon processes of the downloader.

5. Detailed System Design

This section presents the detailed design of the system in UML.

5.1 UML Use Cases Model

Description of the Use cases and Actors

Actors

- **Actor Name:** Client
  Description: The Client here represents the user. The client must be able to fetch files from the internet and download them to an appropriate location.

- **Actor Name:** Browser
  Description: The browser forwards download requests to the downloader providing adequate information required to download the file.

- **Actor Name:** Scheduler
  Description: The scheduler schedules the jobs in the pipeline, facilitates downloading at a later time, optionally followed by a system shutdown.

- **Actor Name:** Daemon
  Description: The daemon is a background process that provides scheduled download and file system monitor services.

Use Cases

- **Use Case Name:** fetch File
  Description: fetches a file from the internet using http.

- **Use Case Name:** fetch file in parts
  Description: fetches a file in parts for efficient bandwidth usage and faster download.

- **Use Case Name:** upload file
  Description: upload files to a server.

- **Use Case Name:** Pipeline jobs
  Description: queue jobs for download.

- **Use Case Name:** Schedule jobs
  Description: fetches a job from the download pipeline and initiates download based on its priority and bandwidth requirements.

- **Use Case Name:** Filter Downloads
  Description: filter files against patterns (regular expressions) for selective download.

- **Use Case Name:** File Open/Save

Description: saves a file into an user specified location or launches an appropriate application to open the file.

- **Use Case Name:** Schedule jobs at a later time
  Description: Schedule jobs at a later time specified by the user, optionally followed by system shutdown.

- **Use Case Name:** Monitor files
  Description: Monitors files specified by user for external changes.

- **Use Case Name:** Operate timer
  Description: start and stop timer, accept time-elapsed events and perform appropriate functions.

- **Use Case Name:** validate xml
  Description: validate an xml document against its schema

- **Use Case Name:** validate WSDL
  Description: validate a WSDL document

- **Use Case Name:** Parse WSDL
  Description: parse a WSDL document

- **Use Case Name:** Pass Parameters
  Description: pass parameters to a web service

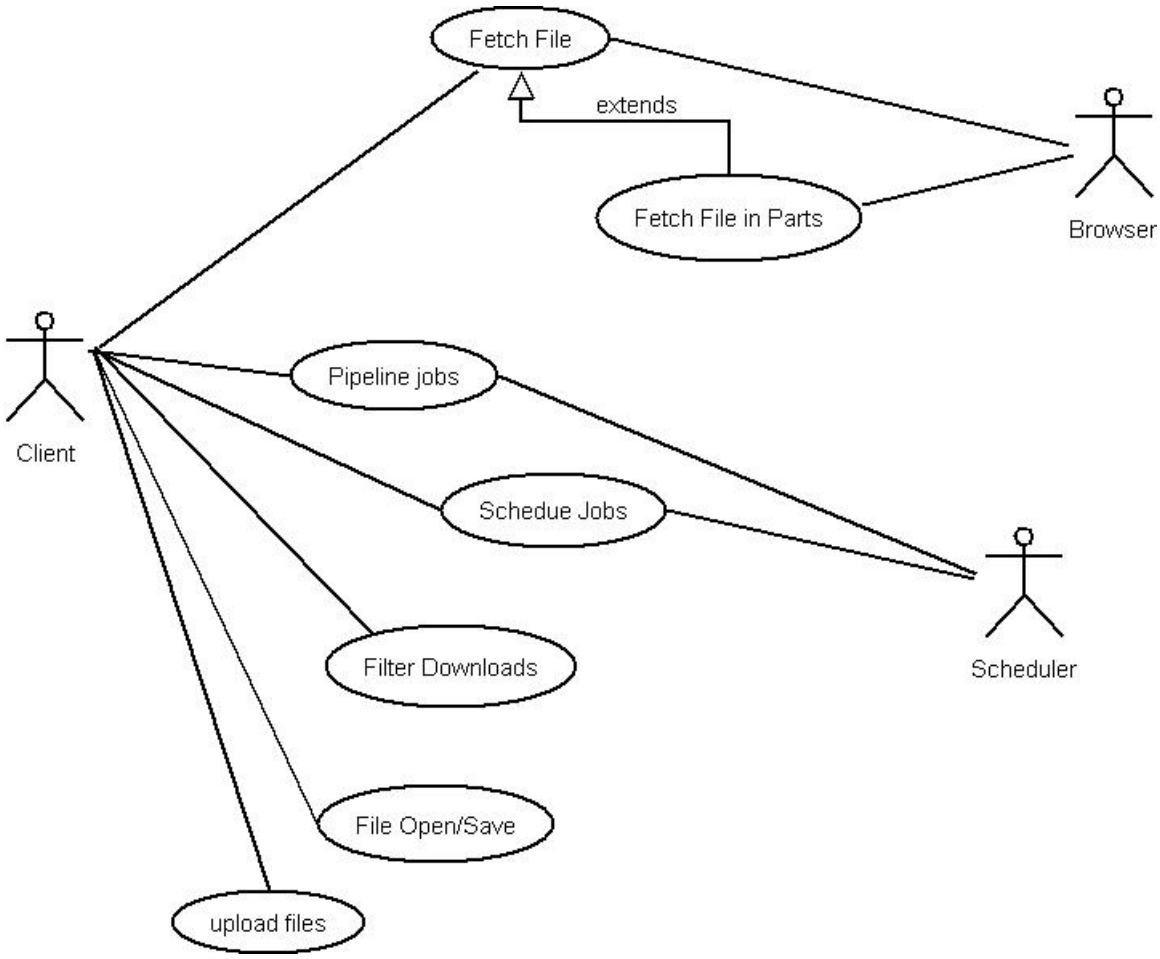

Figure 1 Downloader Package

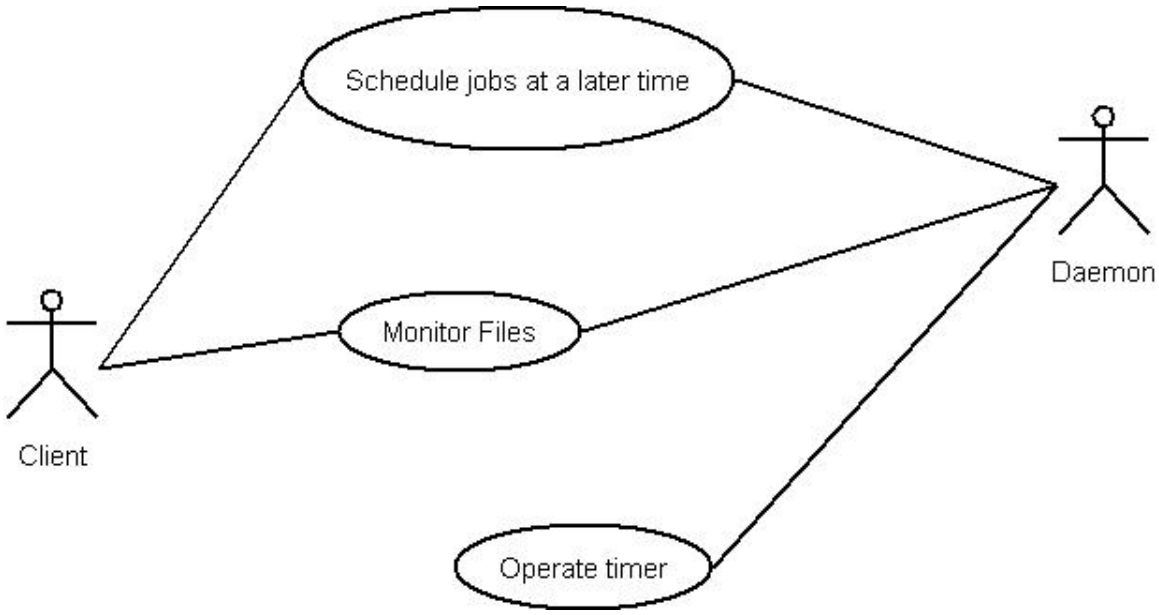

Figure 2 Scheduler Package

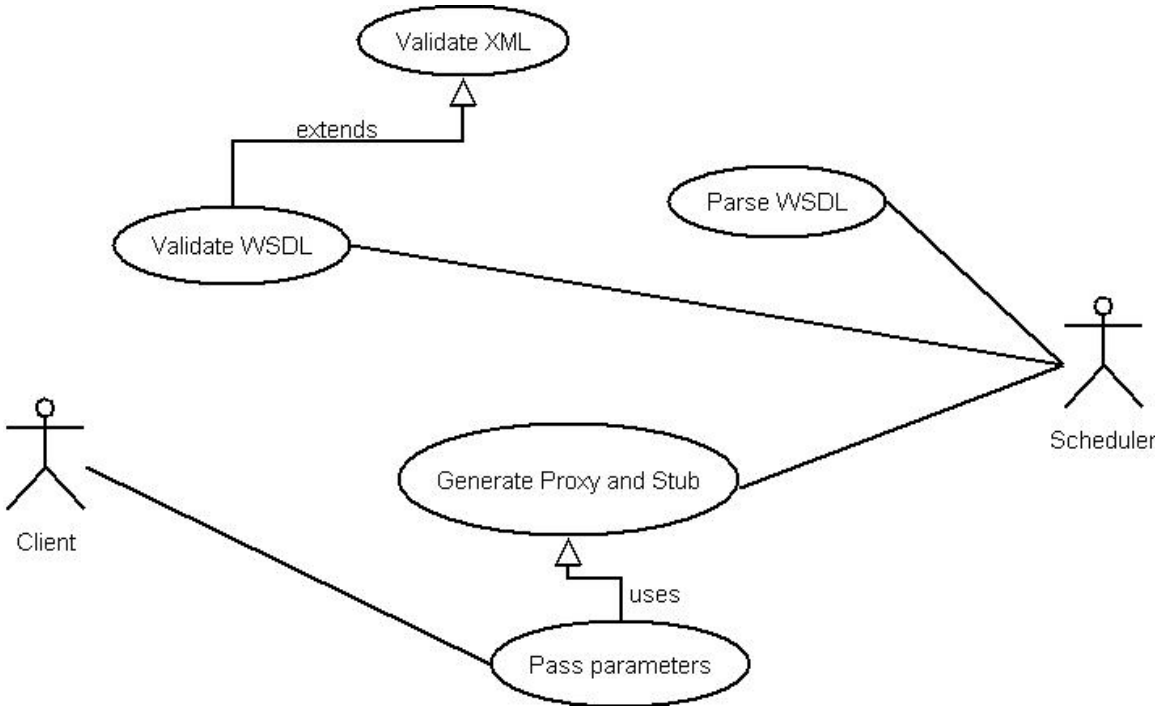

Figure 3 Web Services Package

5.2 UML Interaction Model

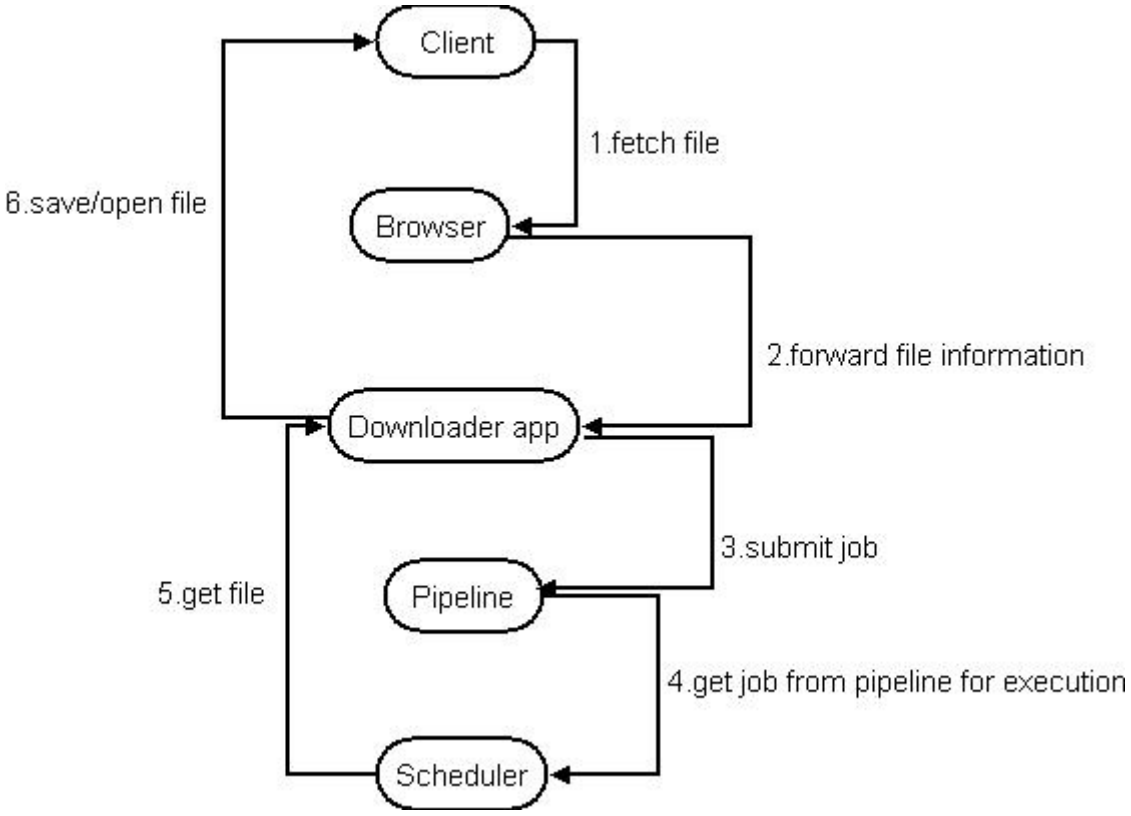

Figure 4 Collaboration diagram for file download

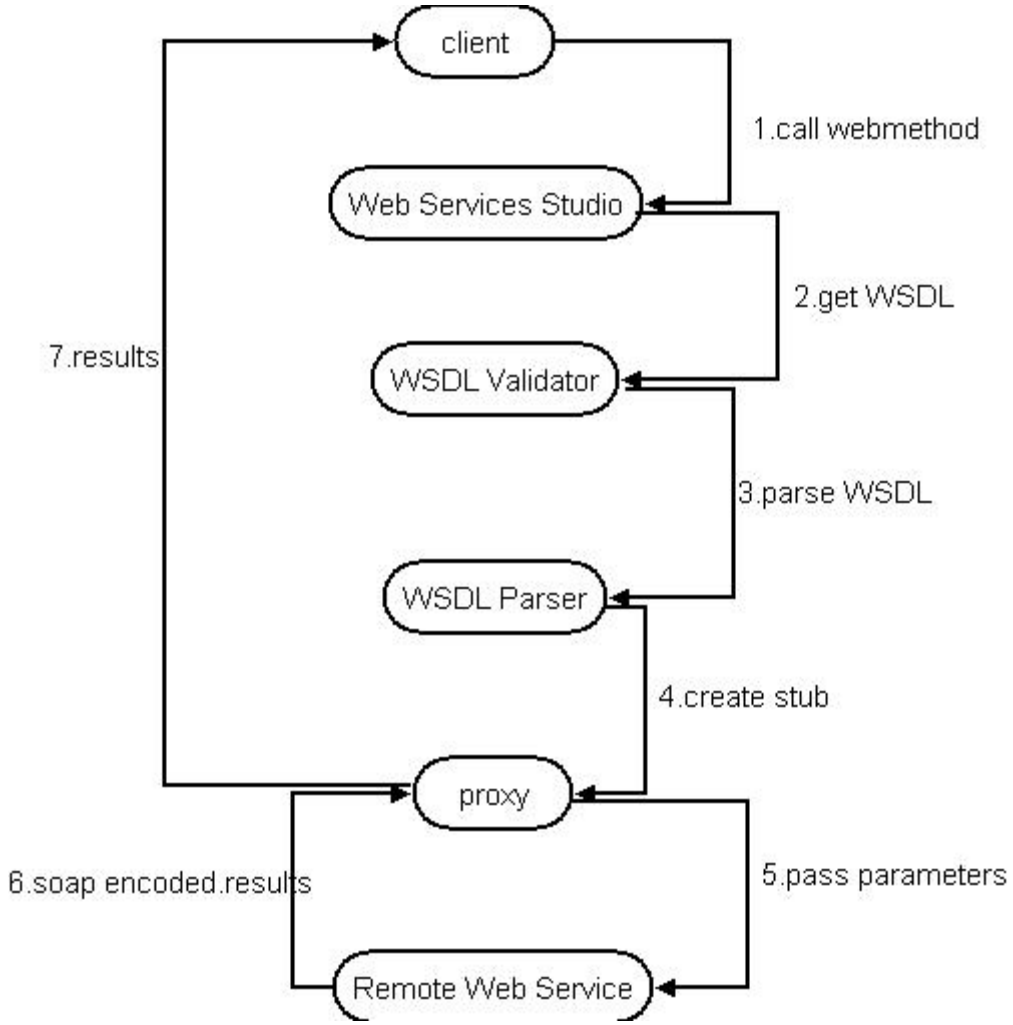

Figure 5 Webservices Collaboration diagram

5.3 Classes and their Responsibilities

Class Name: BasicDownloader
Responsibility: Provides basic downloading services using HTTP.

Class Name: DownloaderWithRanges
Responsibility: Provides multipart downloading services
Inherits: Basicdownloader

Class Name: ScheduledDownloader
Responsibility: Facilitates an user to download at a later time, optionally followed by system shutdown.
Inherits: Basicdownloader

Class Name: Timer
Responsibility: Provides time service

Class Name: MirroredDownloader
Responsibility: Manages sites with mirrors, resuming download from other mirrors if a certain mirror is down.
Inherits: Basicdownloader

Class Name: Scheduler
Responsibility: Fetches download jobs from the pipeline and initiates download based on their priority,size,etc.

Class Name: Pipeline
Responsibility: A Managed queue of submitted jobs for download

Class Name: FileSystemMonitor
Responsibility: Monitors a set of files for external changes and notifies the user(s)

Class Name: XmlValidator
Responsibility: Validates an incoming xml document against its schema

Class Name: WSDLValidator
Responsibility: Validates an incoming WSDL document

Class Name: SoapParser
Responsibility: Parses Soap documents for web services

Class Name: XmlConfigReader
Responsibility: Manages the xml configuration files

Class Name: Security
Responsibility: Provides cryptographic services relevant to a downloader

Class Name: AntiVirusScanner
Responsibility: Scans incoming documents for viruses using existing antivirus softwares.

Class Name: ZipPreview
Responsibility: Provides a quick preview of the contents in a zip archive and allows user to selectively download relevant files from it (archive).
Inherits: Basicdownloader

Class Name: MediaPreview
Responsibility: A thumb preview of a video file without downloading the file
Inherits: Basicdownloader

Class Name: FileSystemVersioner
Responsibility: Scans similar files and saves space by storing only one version of the file, the others are stored as differences that can be applied to the stored one, to get the corresponding files.

5.4 UML Class Diagrams

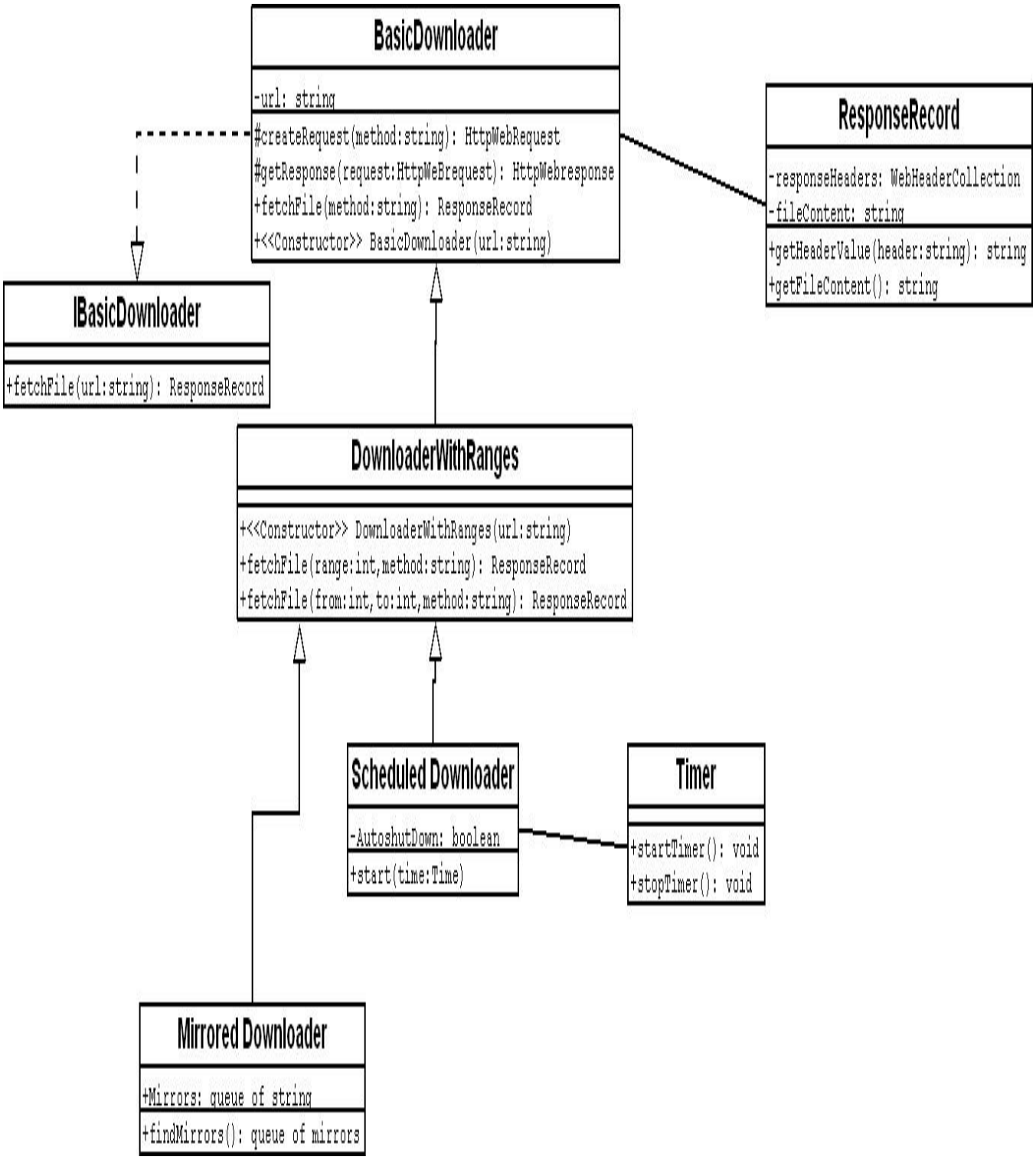

Figure 6 Downloader Package Class diagram

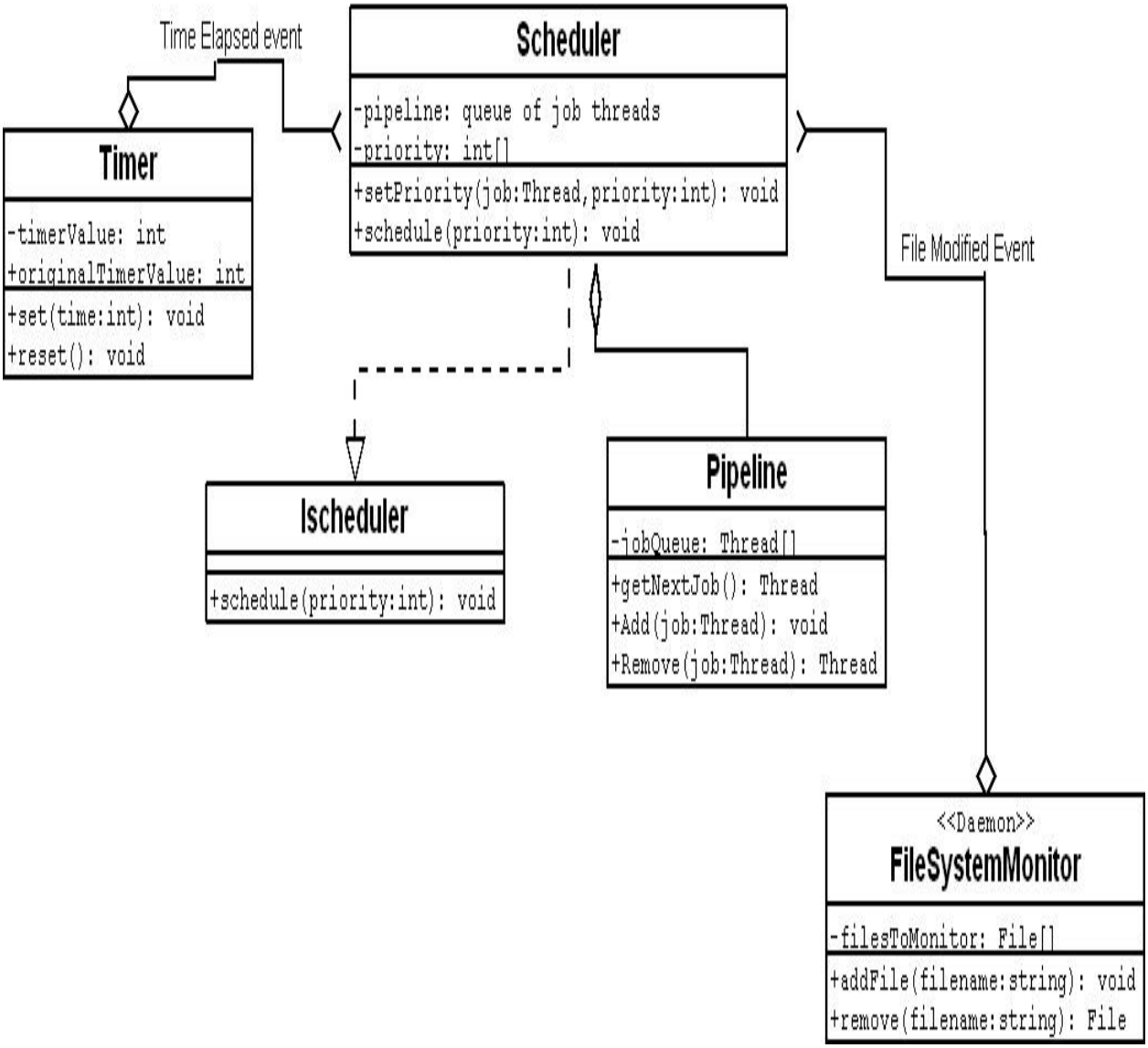

Figure 7 Scheduler Package Class diagram

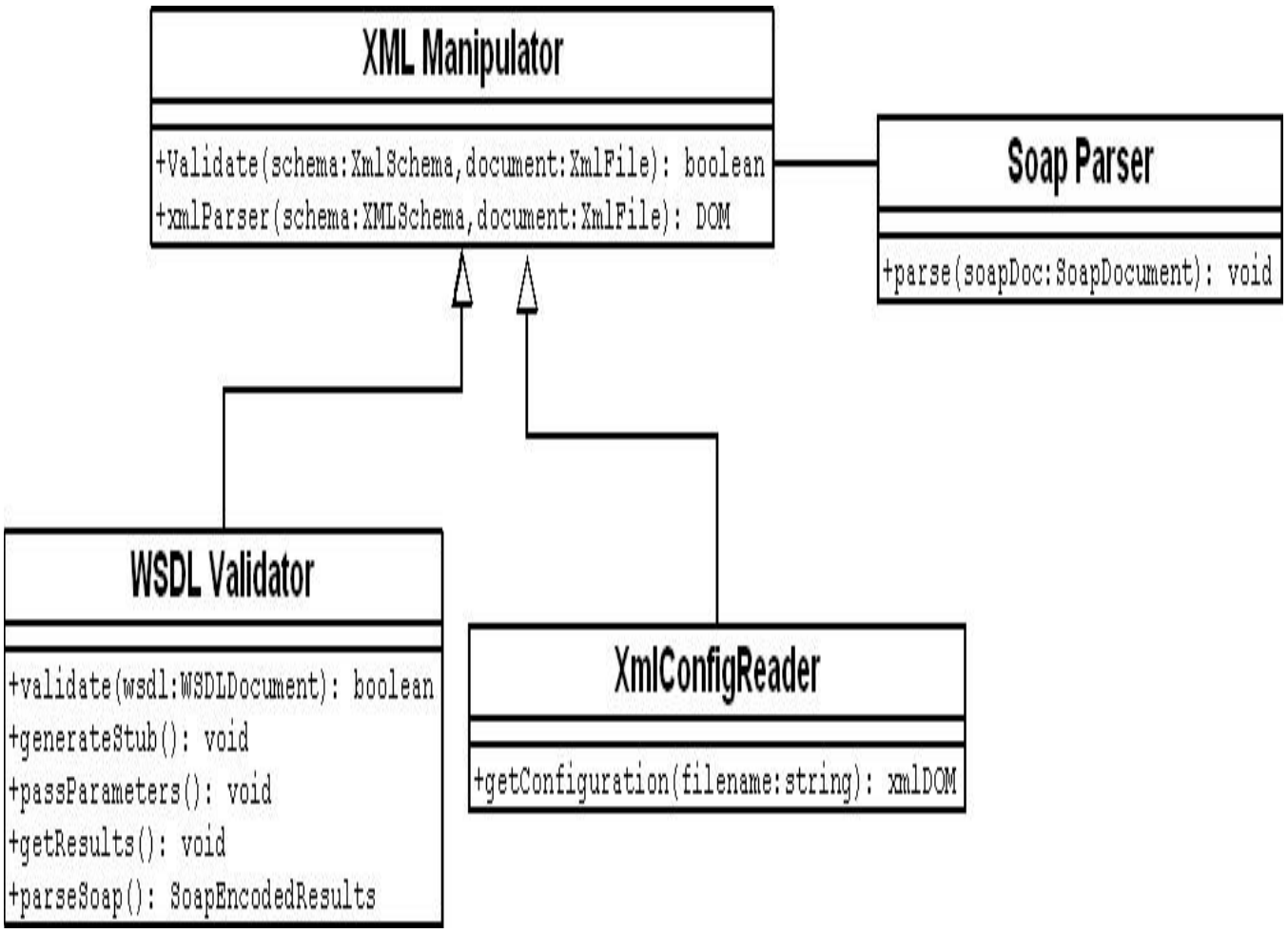

Figure 8 Web services Package Class Diagram

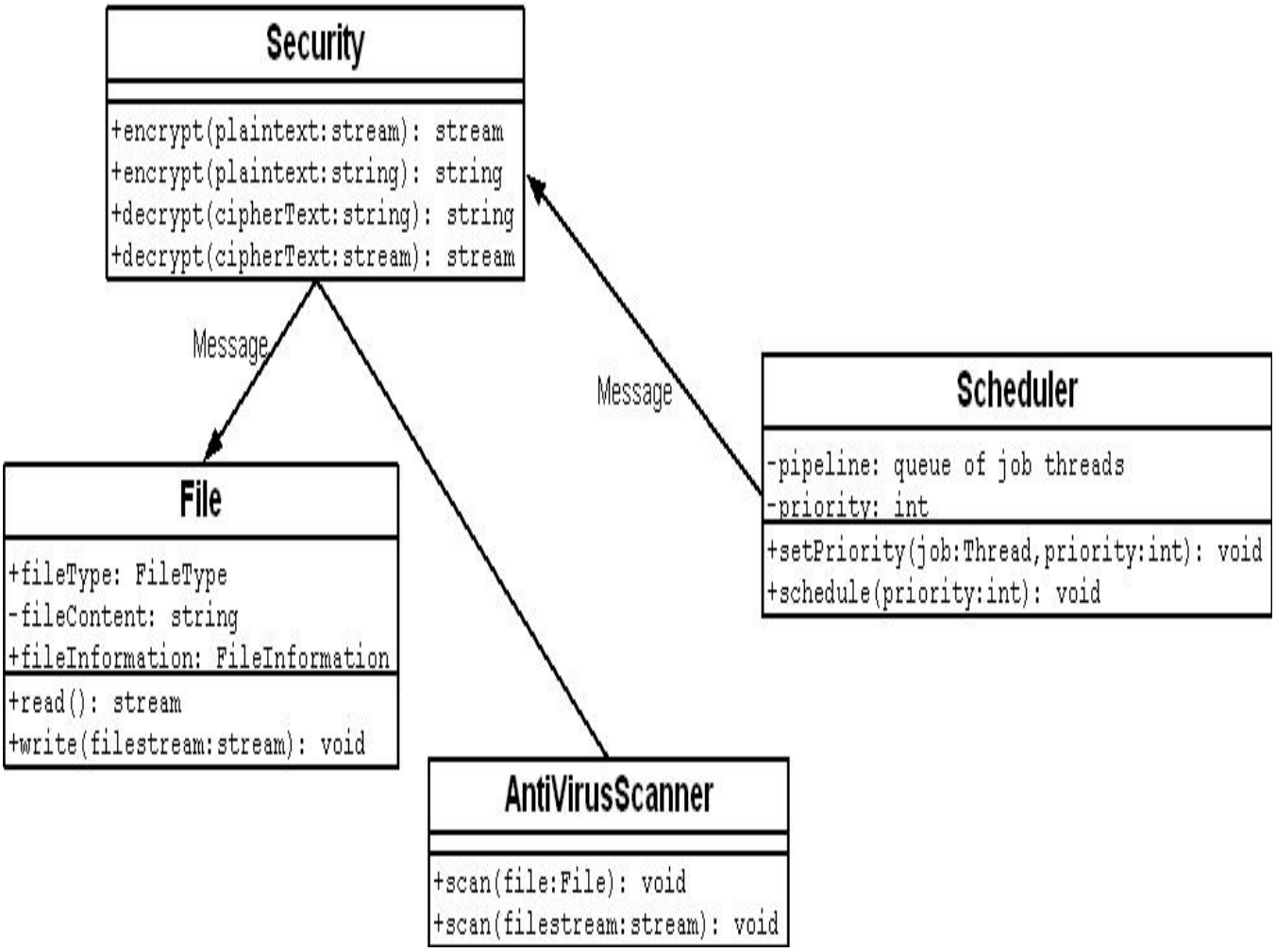

Figure 9 Security Package Class Diagram

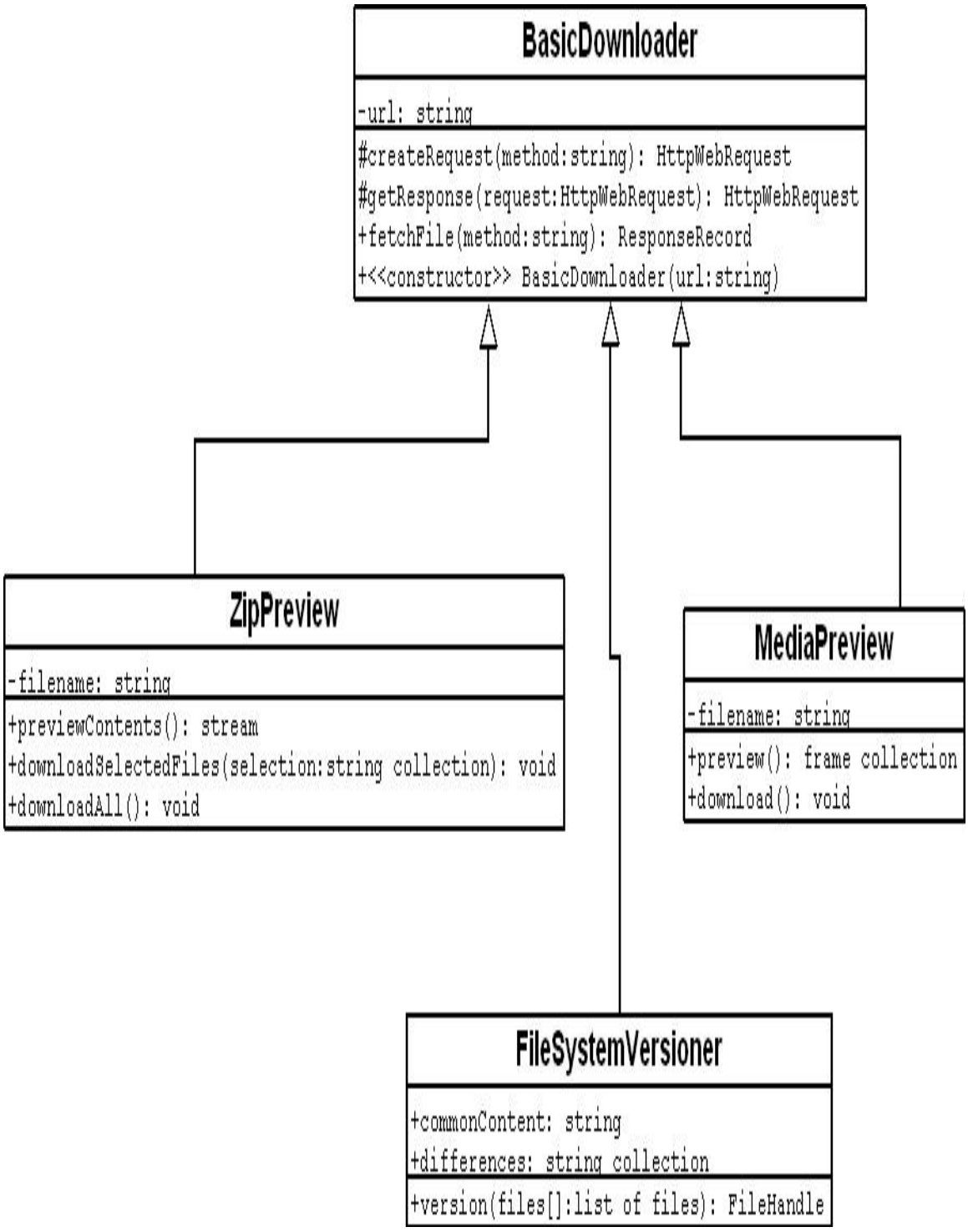

Figure 10 Utilities Package Class Diagram

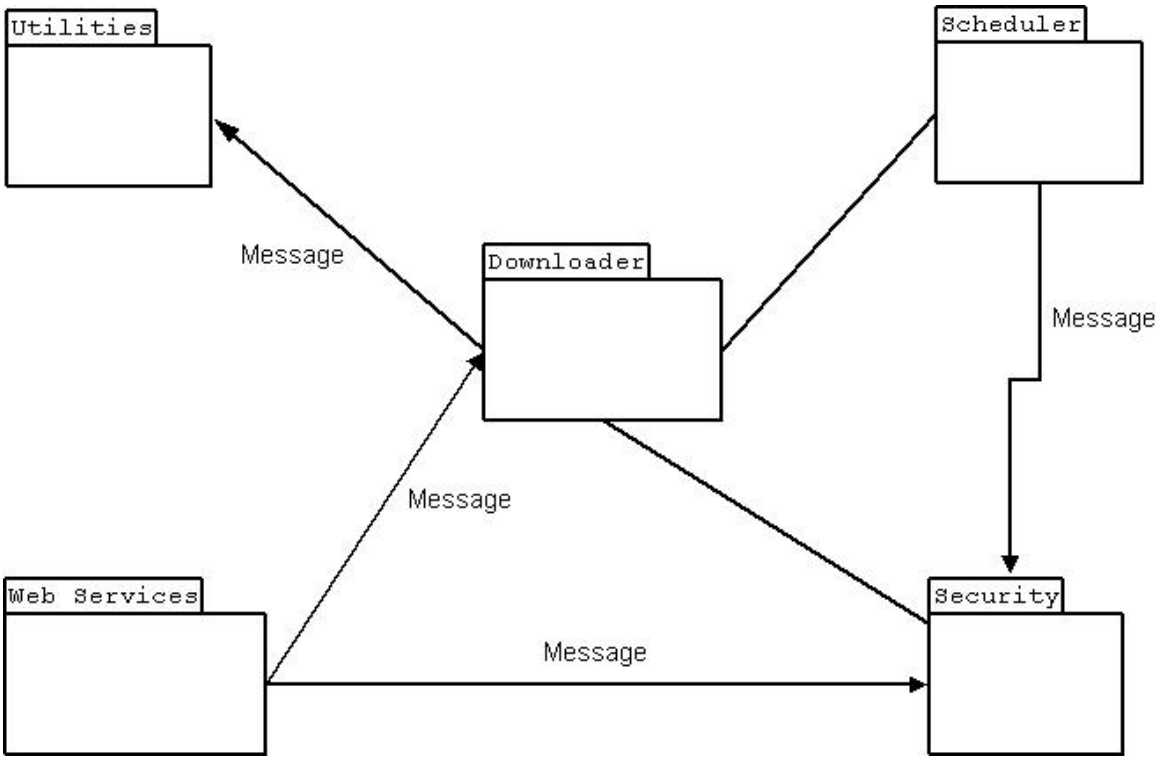

Figure 11 Package Dependencies

6. Glossary

Download Manager

A download manager is a computer program designed to download files from the Internet, unlike a web browser, which is mainly intended to browse webpages on the World Wide Web (with file downloading being of secondary importance).

Web Services

A web service is a collection of protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer.

Proxy

A proxy server is a computer network service which allows clients to make indirect network connections to other network services.

Zip Preview

A preview of the contents in a zip archive.

Media Preview

A thumb preview of a video file.

XML

The Extensible Markup Language (XML) is a W3C-recommended general-purpose markup language for creating special-purpose markup languages. It is a simplified subset of SGML, capable of describing many different kinds of data. Its primary purpose is to facilitate the sharing of data across different systems, particularly systems connected via the Internet.

SOAP

SOAP is a standard for exchanging XML-based messages over a computer network, normally using HTTP. SOAP forms the foundation layer of the web services stack, providing a basic messaging framework that more abstract layers can build on.

WSDL

The Web Services Description Language (WSDL) is an XML format published for describing Web services.

7. Bibliography

http://en.wikipedia.org/wiki/Download_manager

http://gnome.org/projects/gwget/
http://www.internetdownloadmanager.com/
http://www.internetdownloadmanager.com/features.html
http://www.amazesoft.com/
http://www.sun.com/download/sdm/sdm_help.xml
http://www.sun.com/download/sdm/sdm_features.xml
http://www.conceiva.com/products/downloadstudio
http://www.conceiva.com/company/news-2005.asp
http://www.flashgot.net/features
http://www.freedownloadmanager.org/features.htm
http://www.igetter.net/iGetter.html
http://francis.dupont.free.fr/truedownloader/
http://aria-rpm.sourceforge.net/features.php
http://www.speedbit.com/Default_new.asp
http://www.gozilla.com/
http://www.adobe.com/support/products/downloadmanager.html
http://www.smartconverter.com/KazaaDownloadManager-review9190.shtml