Student Updater

Group Members:

Deepti C	Manish Kumar	Meenal Dongle
Prakash	Prashant Kumar	Preethi Murthy
Rashmi	Tushar S	

Contents

Section - 1 **Introduction**

Purpose
Scope

Section - 2 **System Overview**

Section - 3 **System Architecture**

Architectural Design
Decomposition Description
Design Rationale

Section - 4 **Data Design**

Data Description
Data Dictionary

Section - 5 **Component Design**

Section - 6 **Human Interface Design**

Overview of User Interface
Screen Images

Section - 7 **Requirements Matrix**

Introduction

Purpose

The following document includes the reason why the application was chosen to be deployed, the components involved in the system design, architecture and description of the system.

Scope

The application is to be deployed on the Android mobile phone. The effort is towards easing the task of a student in keeping himself updated, from various sources, during his engineering years.

The Goals

Make it easy for students to get updates from sites they commonly access like IISc,NPTEL, ETS, NITK without them having to go to each of them time and again. Updates like internship opportunities in IISc, new videos updated on a subject in NPTEL, GRE dates on the ETS website and news about results being announced on the NITK website are considered under the scope of the project. The user is allowed to like and dislike the updates which appropriately filters the updates next time they are brought.

Users and the benefits offered

Engineering students are the significant users. Immediate updates on information relevant to them and customized to their needs. These updates coming in on a device that is the constant companion of a student, ensures service anywhere well within time.

System Overview

The application is written in Eclipse IDE using the Android SDK. The libraries include html parser libraries.Running the application during the development phase is done on the Android emulator.We intend to finally test the application on an android mobile.

The application will run once every hour, to connect to a website. The html parser constructs the parse tree of the web page. Methods to extract relevant information from this parsing like text, links, email ids etc is provided by the html parser. The 'news' information on the website needs to be extracted. The java code then compares this information with the information, retrieved,the previous time it parsed the web page and updates to the student, the changed information.

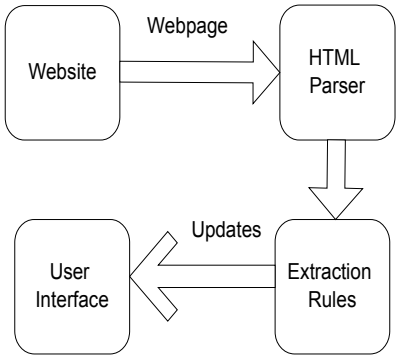
This changed information is reported to the user in the form of a small icon on his home page. If he wishes to see the updates, he can click on them to be taken to another page where the updates is listed.This is shown in the architectural design below.

System Architecture

Architectural Design

The extraction rules determine what exactly is needed from the parsed information. Since the 'news' is important for us, we need to identify rules which will extract this information from the parse tree.

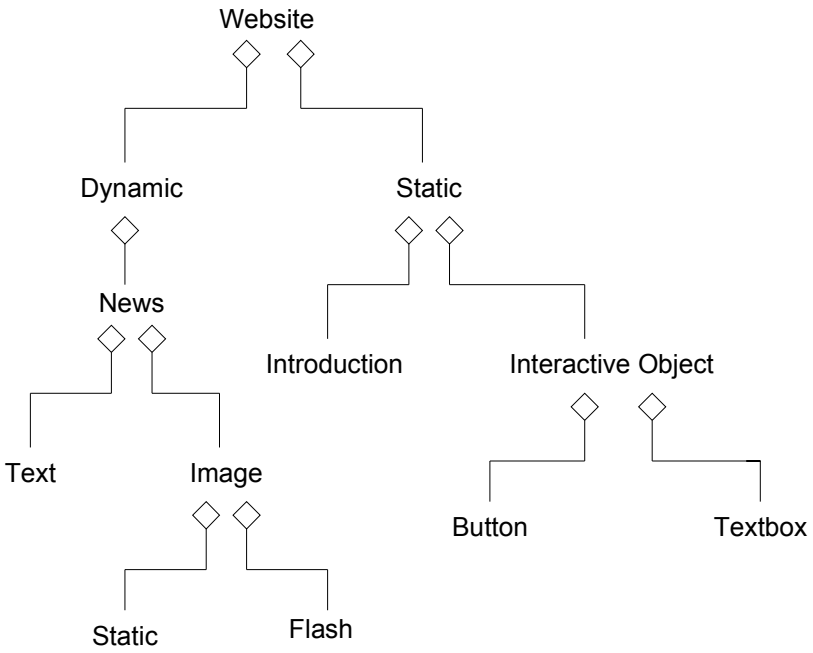
The updates are stored in a file before they are shown to the user. There is a static file with which we compare the extracted information to assess if anything new is updated. This new information is appended to the beginning of the same file.



Decomposition Description

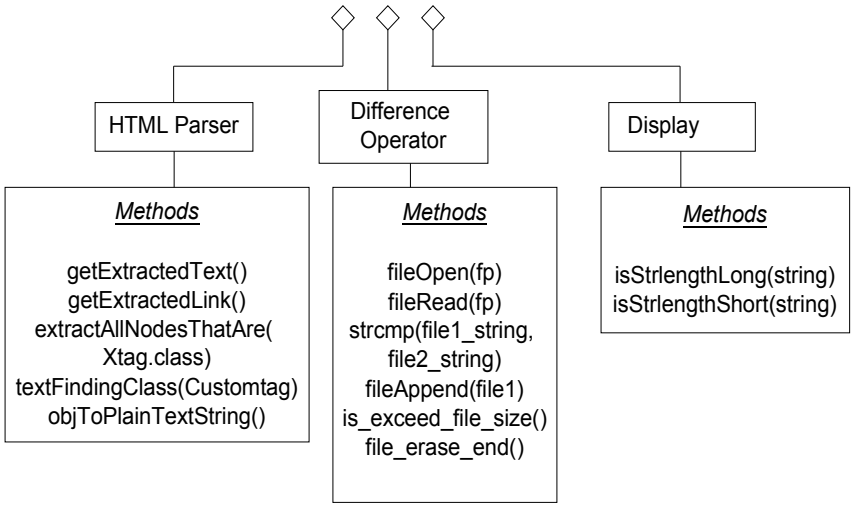
Here we identify the objects involved in our system.The position of the 'news' varies, the news maybe a flash object, an image or simple text. To identify the extraction rules for each website, our first object of study is a website.

Aggregation model

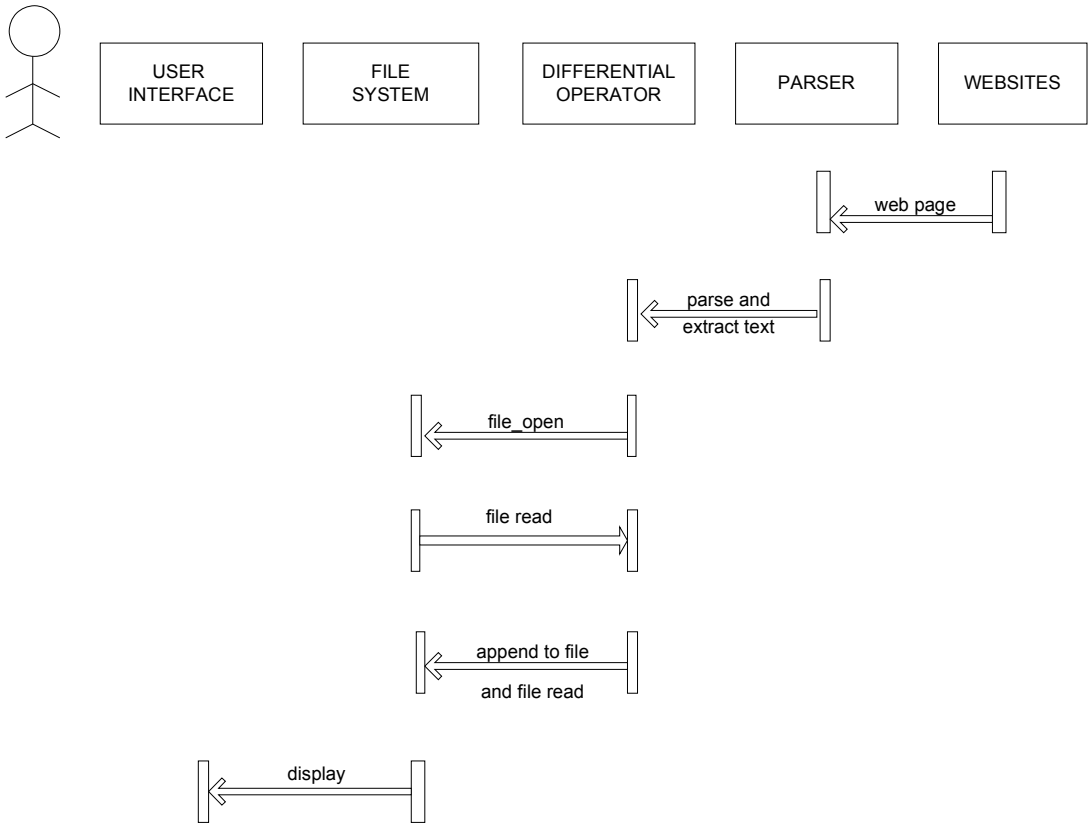


The below model (Next Page) talks about the modules involved in our application. The extraction rules mentioned in the architectural design above is jointly implemented by the difference operator and the html parser. Display is equivalent to the UI mentioned in the architectural design.

Application



Behavioral Model



Design Rationale

The websites chosen for the application is restricted as of now to NPTEL, IISc, NITK and the GRE. The reason as was mentioned above is that they are the websites from which an engineering student wants updates as quickly as possible. The main page of these websites are made of html, javascript. Sometimes flash objects are included.

Although the HTML parser has flexible parsing capabilities, extracting the 'news' from it depends to a certain extent on the content around it. The selected websites are those whose contents do not change that often. Thus the extraction rules need to take care of less probable minor changes.

Risks: When the file, which stores the the updates exceed in size, we allow the last few contents to be automatically deleted. This will give rise to a problem if the user wishes to retain that update. To over come this we provide the user the LIKE option which enables him to choose what he wishes to retain. This feature, we will include once the presented idea is implemented.

Data Design

Data Description

A file is maintained to store the updates (File A). Each time parsing is done and newer updates are found, they are written into another file(File B). File B is checked to see if it has anything new to add. If so it is appended to the beginning of File A.

Contents of File A is displayed to the user. But the problem arises when the file size grows. Therefore after a certain size, the last contents of the File A is deleted, so as to fit the screen comfortably.

Data Dictionary

Difference Operator

Type: Object
Attribute: fp1,fp2,file1_string,file2_string

Methods:
1. fileOpen(fp)
2. fileRead(fp)
3. strcmp(file1_string,file2_string)
4. fileAppend(file1)

Descriptions:
1. fp1,fp2 are the file descriptors which represent the file to be compared
2. with and the file which is being compared
3. fp1 refers to the file which has all the updates so far
4. fp2 refers to the file which has the recently parsed updates
5. fileOpen() is used to open each of the above files while comparing them
6. fileRead() is used to read the contents of the files
7. The strings retrieved from both files after reading are compared using strcmp()
8. The new update is added at the beginning of file described by fp1, using the function fileAppend()

Display

Type: Object
Attribute:string

Methods:
1. isStrlengthLong(string):
2. isStrlengthShort(string):

Descriptions:
1. isStrlenghLong(): is used to see if the update is too long. If it is, another method displays 'click here for more'
2. isStrlenghShort(): is used to see if the update is short. if so, another method displays the entire update.

FS

Type: Data storage entity

It stores the file which has the more recent updates, and the file which has the recently parsed update.

HTML Parser

Type: Object
Attribute: url

Methods:
1. getExtractedText()
2. getExtractedLink()
3. extractAllNodesThatAre(Xtag.class)
4. objToPlainTextString()
5. textFindingClass(CustomTag)

Descriptions:
1. getExtractedText() extracts only the text content of the parsed website.
2. getExtractedLink() extracts just the hyperlinks of the parsed website.
3. extractAllNodesThatAre() extracts the tags X and passes the tags to objToPlainTextString() to extract the text associated with that tag.
4. textFindingClass() extracts the tags which arrive in the order mentioned in custom tags and pass it on to objToPlainTextString() to extract the text associated with that tag.

Website

Type: External Entity

The 'news' which we look at to be displayed to the user is simple text in all the sites mentioned in the beginning, NPTEL, IISc, NITK, GRE. The variation is in where and after what the news appears. This is brought in for the HTML parser.

Component Design

The following section describes briefly the functions of each of the objects mentioned above:

1. **HTML Parser:** It has the capability to extract exactly what is required from a web page, apart from simply constructing the parse tree for the web page. Therefore its output is the 'news' content.

2. **Difference Operator:** takes in this news content which is stored in a file, compares with the file containing the previous updates, appends the changed content to this file. If the file contents exceed a limit, it deletes the last few contents.

3. **Display:** reads the content of the file containing updates and decides how to display it. It either displays the entire update or says 'see here for more'.

Human Interface Design

Overview of Human Interface

- The user sees an icon on top of the home page where time, charge, signal are displayed if new updates are found
- If he decides to click on that icon, he is taken to a page where there are 4 icons for each of the four websites
- Only those website icons are enabled which have updates for them, else they are disabled
- The user decides which updates he wishes to see
- On clicking on the icon, he is taken to the main page of the updates

- If the updates are too long, a 'see here for more' is displayed which hides the other updates and displays the long update on the same page
- We wish to extend the usability by allowing the user to like/dislike an update so that we can alert him only for updates that he is interested in.

Screen Image