

IT 611 Course Project

Site Feeds Aggregator

J. Ramanand

{05329402}

17 October 2005

Space

- * Websites publish changes to their content by means of “*Site Feeds*”
- * “*Aggregators*” monitor these feeds for content and changes
- * A site feed is usually an XML file
- * Popular specifications: RSS (Rich Site Summary/Really Simple Syndication), Atom
- * e.g:

```
<?xml version='1.0' encoding='utf-8' ?>
<rss version="0.91">
<channel>
  <title>Cricket News</title>
  <link>http://mycroft.it.iitb.ac.in/ram.rss</link>
  <description>Cricket Information Services</description>
  <language>en</language>
<item>
<item>
  <title>England slump to defeat versus Pakistan</title>
  <link>http://mycroft.it.iitb.ac.in?id=5</link>
</item>
</channel>
</rss>
```

Domain Analysis

{ Important Concepts }

- * The idea of an “*Aggregator*”
- * Aggregators have “*Feeds*”
- * Feeds have “*Items*” and have a “*Type*” { Atom/RSS }
- * XML parsing capability
- * Ability to get files from a remote web server

Key Basic Requirements

- * Contain multiple feeds
- * Support popular XML Feed specifications: RSS, Atom
- * UI for Adding, Managing, Viewing Feeds
- * Fetch feed files hosted on a Web Server
- * Determine when additional “items” have been added to a Feed when asked to

Key Use Cases - I

Use Case 1: View List of Feeds

1. User: Launches application
2. System: Presents Menu
3. User: Selects Option to View all Feeds
4. System: Retrieves a list of all registered Feeds and displays them

Use Case 2: View Items of a Feed

1. User: Launches application
2. System: Presents Menu
3. User: Selects Option to View all Feeds
4. System: Retrieves a list of all registered Feeds and displays them
5. System: Presents Menu for selection of a Feed
6. User: Selects the ID of a feed whose items are to be shown
7. System: Shows all items belonging to the selected Feed

Key Use Cases - II

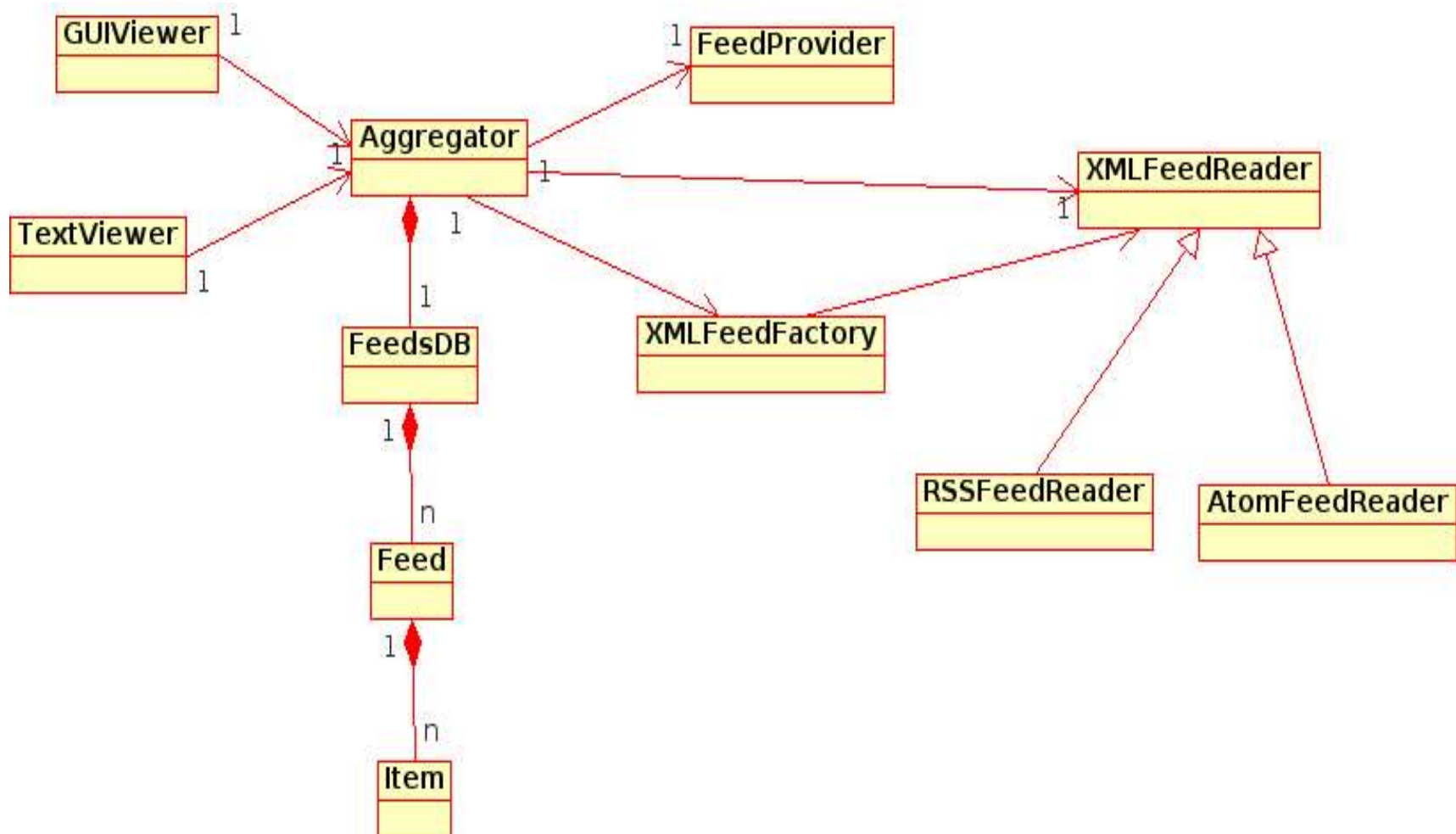
Use Case 3: Add a New Feed

1. User: Launches Application
2. System: Presents Menu
3. User: Selects Option to Add a New Feed
4. System: Queries for the URL's feed
5. User: Enters a valid URL for the Feed
6. System: Downloads the feed, processes it and adds the feed to the System

Use Case 4: Monitor Feeds

1. User: Launches Application
2. System: Presents Menu
3. User: Selects Option to Monitor Feeds
4. System: Checks each feed and announces any additions to the feeds

Class Design



Role Stereotypes

1. Aggregator: Controller, Service Provider, Coordinator
2. FeedDB: Information Holder
3. Feed: Information Holder
4. Item: Information Holder
5. TextViewer: Interfacer
6. XMLFeedReader/RSSXMLReader/AtomXMLReader:
Service Provider
7. XMLFeedFactory: Factory Object
7. FeedProvider: Service Provider

OO Concepts applied

- * Modular separation of responsibilities (XML parsing, Feed Procurement, Representation of Feeds, Items etc)
- * Appropriate Inheritance: XMLFeedReader & AtomXMLReader/RSSXMLReader
- * Design Patterns:
 - Singleton: Aggregator object, Feed Provider
 - Factory: XMLFeedFactory
- * Error handling using Exceptions

Infrastructure

- * Java 5.0 on Linux
- * JUnit for Testing
- * Eclipse for Development
- * DOM XML Parser
- * Java classes for HTTP

Testing

- * High level Test Cases Identified during Requirements Gathering
- * Key Classes – Test Cases written before Implementation
- * Basic Test Plan
- * Test Cases:
 - Partial coverage using Junit
 - Manual testing for the rest

Limitations and Enhancements

- * User driven monitoring – Add concurrent monitoring and callbacks
- * GUI more suited for such apps – Add a GUIViewer and use an Observer Pattern
- * Assumptions on feed formats – Handle better variations in the formats