ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

EiffelRSS

Project description

Michael Käser <kaeserm@student.ethz.ch>
Martin Luder <luderm@student.ethz.ch>
Thomas Weibel <weibelt@student.ethz.ch>

inf | Informatik
Computer Science

Abstract

EiffelRSS is an Eiffel library to read and write RSS. The goal is to provide the Eiffel development community with an easy to use and well structured API for RSS.

The distribution also contains a RSS newsfeed reader written with EiffelVision and EiffelRSS.

1 Library

The library of EiffelRSS consists of several different clusters.

1.1 ADT

`ADT` contains the deferred classes `SORTABLE` and `ORDER_RELATION` which can be used to implement sortable structures.

`SORTABLE_TWO_WAY_LIST` inherits from `SORTABLE` and `TWO_WAY_LINKED_LIST` to implement a sortable doubly-linked list.

See figure 1 for an overview of the cluster.

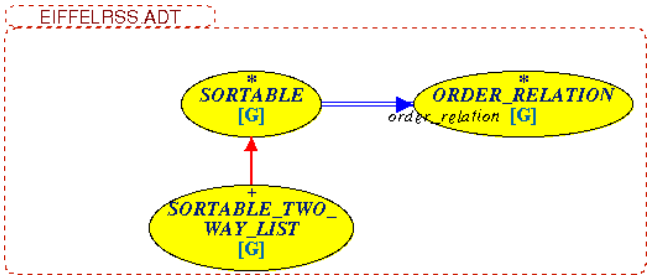

Figure 1: BON diagram of cluster `ADT`

1.2 FETCH

`FETCH` contains classes which can fetch data from a source address to a local `STRING` using various services. The class `FETCH` provides a simple interface for the `DATA_RESOURCE` class in EiffelNet.

A valid source address has the following format: `service://address`.

Supported services are: file, http, ftp.

See figure 2 for an overview of the cluster.

1.3 LOGFILE

`LOGFILE` represents a file which can be used for logging messages during the program execution.

Each message has its own priority (a positive integer value) and each logfile a certain user defined threshold. If the priority of the message is greater or equal than the threshold, it gets written to the log file together with a timestamp.

See figure 3 for an overview of the class.

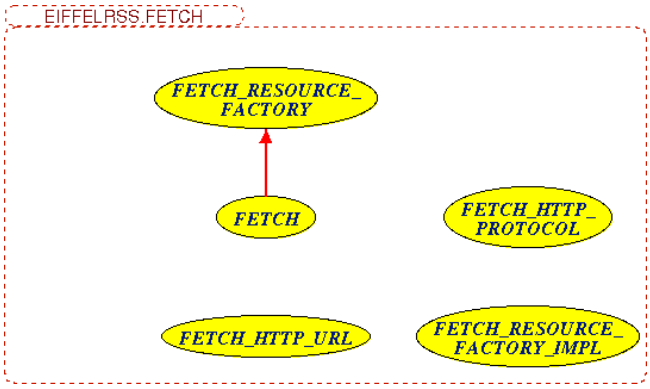

Figure 2: BON diagram of cluster `FETCH`

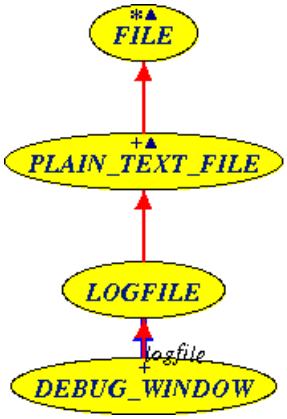

Figure 3: BON diagram of class `LOGFILE`

1.4 PROPERTIES

`PROPERTIES` represents a persistent set of properties. The properties can be saved to a file or loaded from a file.

Each key and its corresponding value in the property list is a string.

A property list can contain another property list as its default. This default property list is searched if the property key is not found in the original property list.

`PROPERTIES` is similar to the `java.util.Properties` class. The main difference is, that `PROPERTIES` doesn't support keys and values separated by whitespace only. It always expects `:` or `=` as a separator between key and value.

See figure 4 for an overview of the cluster.

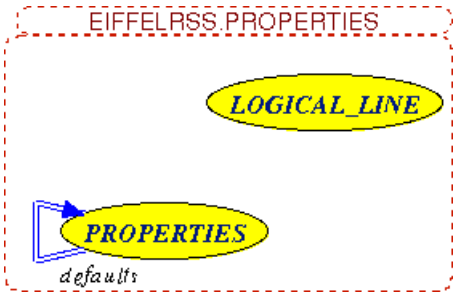

Figure 4: BON diagram of cluster `PROPERTIES`

1.5 SYNDICATION

`SYNDICATION` is the main cluster of EiffelRSS with a feed object model, and classes to load and write feeds. It is divided into three subclusters.

See figure 5 for an overview of the cluster.

1.5.1 INTERFACE

`INTERFACE` is the sub-cluster of syndication with all the classes a developer needs to use the library. There are classes to read into and write from a `FEED`, a `FEED_MANAGER` to administrate a list of `FEED`s, and a factory class which makes it easy to create all necessary objects.

See figure 6 for an overview of the cluster.

1.5.2 FEED

`FEED` is the central datastructure of EiffelRSS. It defines an abstract syndication feed.

See figure 7 for an overview of the cluster.

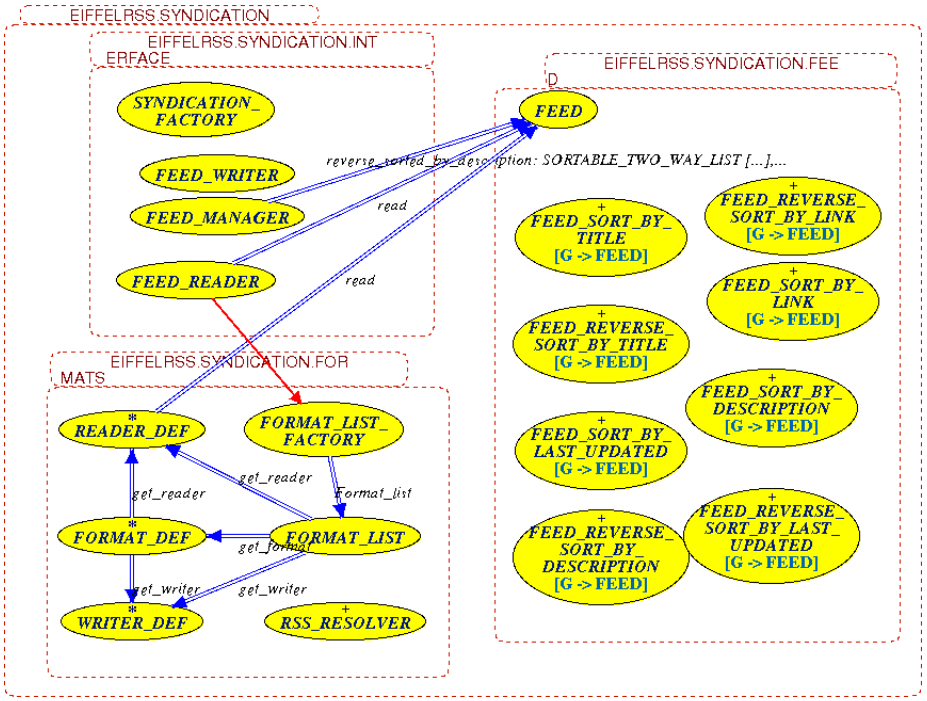

Figure 5: BON diagram of cluster `SYNDICATION`

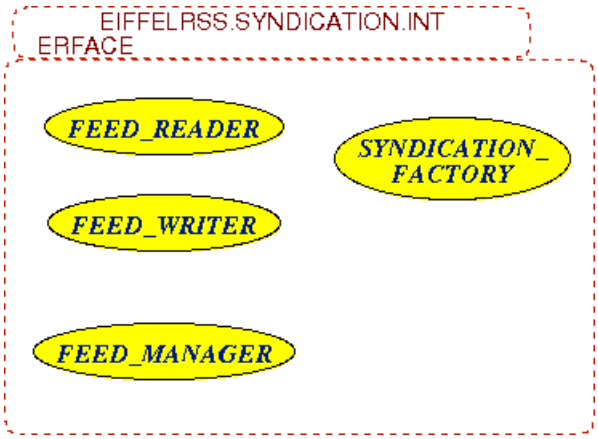

Figure 6: BON diagram of cluster `INTERFACE`

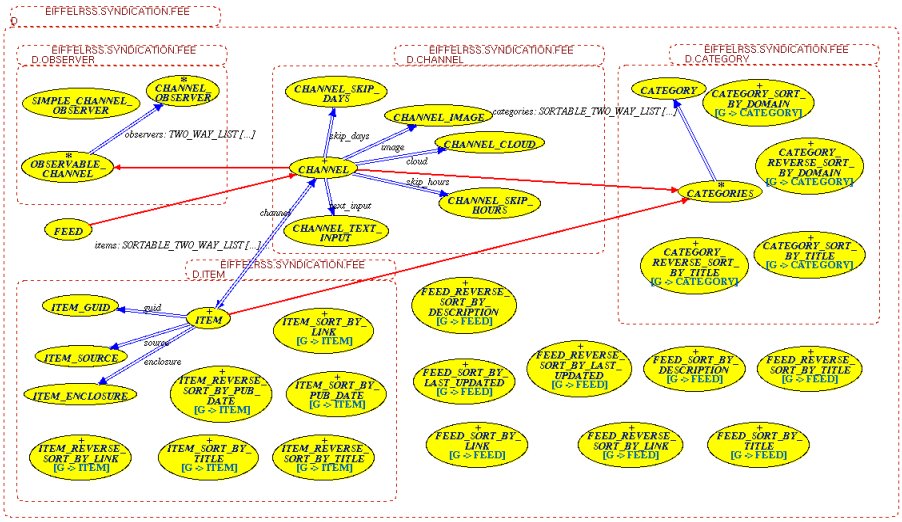

Figure 7: BON diagram of cluster `FEED`

1.5.3 FORMATS

`FORMATS` defines the different syndication formats.

See figure 8 for an overview of the cluster.

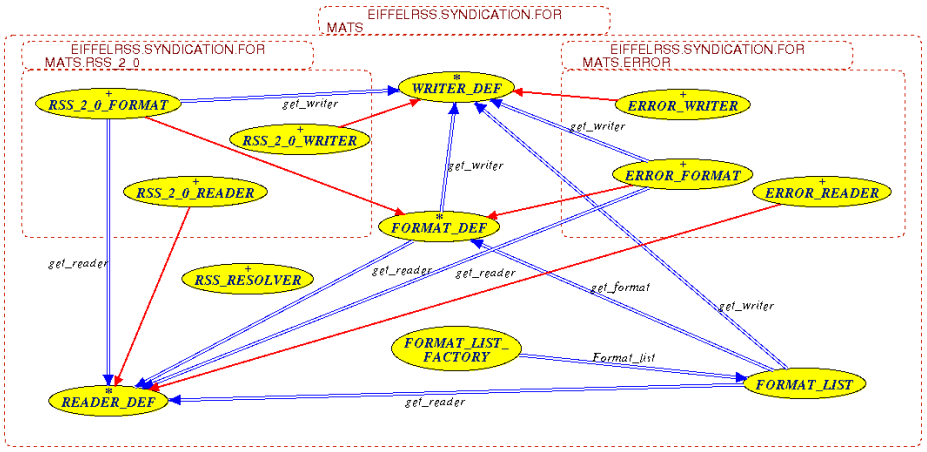

Figure 8: BON diagram of cluster `FORMATS`

2 Newsreader

Newsreader is a simple GUI, with which you can read your RSS feeds. In addition to the GUI it also has a command line user interface.

It is possible to add custom feeds and open news in your Internet browser.

Figure 9 shows a screenshot of the graphical user interface.

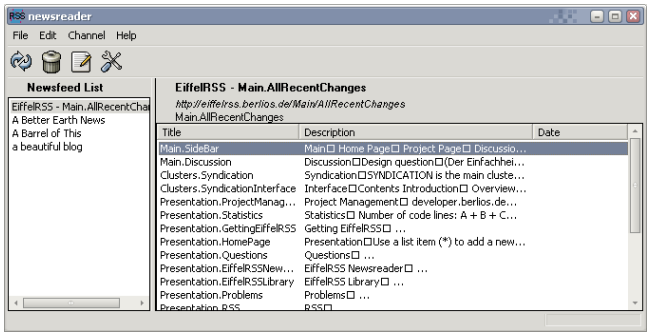

Figure 9: Screenshot of the graphical user interface of Newsreader

3 Extensibility

EiffelRSS is designed with extensibility in mind.

Initially EiffelRSS will only support reading of RSS 2.0. But the library is designed to be easily extensible by other RSS readers. Even non-RSS syndication formats like Atom can be parsed into EiffelRSS' datamodel. It is also feasible to add writing support and to write any syndication format (even non-XML). Because EiffelRSS uses an abstract intermediate representation of newsfeed data, one can also convert from one format to another.

Another possibility of extension would be the newsfeed reader written with EiffelVision. This application could be extended to a full fledged newsfeed reader.

4 Who does what?

4.1 All project members

- Writing documentation
- Testing

4.2 Michael Käser

- The clusters `LOGFILE` and `FETCH`
- An abstraction for readers and writers which translate between XML files and the `FEED` object structure
- An implementation of RSS 2.0

4.3 Martin Luder

- Newsreader with graphical and command line user interface

4.4 Thomas Weibel

- The clusters `ADT` and `PROPERTIES`
- `FEED` data model