
RSS 구독 시스템을 위한 패키지의 설계

이동규* · 김윤호**

A Package Design for RSS Reader Systems

Dong-Kyu Lee* · Yun-Ho Kim**

요 약

RSS 구독기는 웹에서 배포하고 많고 자주 변하는 RSS 포맷의 콘텐츠를 수집하여 구독자가 쉽게 구독할 수 있도록 도와주는 소프트웨어 애플리케이션이다. 본 논문에서는 RSS 구독기의 구현을 위하여 RUP 객체지향 설계 방법론에 의거한 체계적인 소프트웨어 설계를 제시하고자 한다. RSS 구독기의 구현을 위한 요구사항을 기초로 하여 유스 케이스를 추출하고, 도메인 컨셉 모델을 제시한다. 이들을 바탕으로 사용자와 RSS 구독기간에 발생하는 이벤트와 논리적 아키텍처를 추출하여 RSS 구독기의 클래스와 패키지를 설계 한다. 아직 초기에 있는 RSS 구독기에 대한 체계적인 설계 방법이 제시됨으로서, 이에 기반 한 다양한 특성의 기능을 가진 RSS 구독기의 손쉬운 제작이 가능하다.

ABSTRACT

The RSS Reader is a software application that assist users easily to read and collect contents which are frequently changeable syndicated contents of RSS format that published on the web. This paper presents a formula software design based on RUP object-oriented design method for implementing the RSS Reader. We extract a use case based on requirements for implement of the RSS Reader and propose a domain concept model by. Using above artifacts, we extract event to generate between users and, the RSS Reader and logical architecture. So, we design classes and packages of the RSS Reader. As proposed a formula software design about the RSS Reader in early stage, it can make the RSS Reader with a variety of characters easily based on these.

키워드

RSS READER, UML, RUP, OOA/D

I. 서 론

팀 버너스 리(Tim Berners-Lee)에 의해 WWW(World Wide Web)이 제안된 이래, 현재 기업과 언론사는 물론, 개인에 이르기까지 서로가 가지고 있는 웹 콘텐츠를 자유롭게 공유해 왔다. 하지만 이러한 웹 콘텐츠가 폭발적으로 증가하면서, 필요한 콘텐츠를 찾지 못하거나, 찾

라도 계속 업데이트되는 내용을 확인하는데 있어서 많은 불편함을 겪고 있다. 이러한 문제를 해결하고자 하는 목적으로 RSS(Really Simple Syndication)[1][2]가 제시되었으며, 이것은 웹 콘텐츠를 규격화하여 콘텐츠 제공자는 손쉽게 배포하고 사용자는 편하게 업데이트된 내용을 획득할 수 있도록 하기 위한 하나의 해결책이다.

RSS는 자주 업데이트 되는 웹 콘텐츠를 쉽게 저작하고

* 안동대학교 컴퓨터공학과 석사과정
** 안동대학교 전자정보산업학부 부교수

배포할 수 있도록 해주는 XML기반 포맷이다[3]. 따라서, 사용자가 RSS 문서를 구독하기 위해서는 RSS 피드 형태로 제공되는 서버의 정보를 해석하여 원하는 콘텐츠를 구독할 수 있도록 하는 RSS 구독기가 있어야 한다. 현재 일반 사용자에게 제공되는 RSS 구독기의 형태는 사용자의 시스템에 설치하여 사용하는 데스크탑 기반의 RSS 구독기와 웹 브라우저를 사용하는 웹 기반 RSS 구독기의 두 종류가 있다[4][5][6][7][8][9].

최근에 여러 형태의 RSS 구독기가 개발되어 발표되고 있지만, 아직 초기단계이다[3]. 따라서, 본 논문에서는 RSS 구독기를 구현하기 위한 목적으로, RUP (Raional Unified Process) 방법론에 기반하여 객체지향적으로 설계된 패키지를 제시하고자 한다. 즉, 데스크탑 기반의 RSS 구독기를 구현하기 위하여 RUP 방법론에 입각하여 요구사항을 분석하고 이에 기초하여 RSS 구독을 위한 패키지를 식별하고 설계한다.

II. RSS 구독기 분석

객체지향 분석 및 설계(Object-Oriented Analysis and Design)의 대표적인 방법론인 RUP는 UP(Unified Process)를 상세하게 정제한 것이다[10]. RUP의 프로세스와 각 단계별 산출물들의 관계를 그림 1에서 보인다. 즉, 요구사항 분석에서는 기능적 요구사항과 비기능적 요구사항을 추출하고, 추출한 결과에서 사용자 애플리케이션을 사용하는 흐름이나 시나리오를 식별하게 되는데, 이 흐름을 UML[11]의 유스케이스를 사용하여 작성한다. 그리고 객체지향 분석과정에서는 유스케이스를 기초로 객체 또는 개념의 관점에서 문제 도메인을 설정하고, 이들에게 주목할 만한 개념, 속성, 연관 관계 등을 식별해낸다. 그리고 문제 도메인의 개념이나 객체를 도메인 모델(Domain Model)로 표현하며, 이는 시스템의 정적인 관점에서의 분석을 의미하게 되므로 클래스 다이어그램으로 표현한다. 또한, 객체지향 설계에서는 객체의 책임이나 다른 객체와의 협동을 식별하여 시스템의 동작적인 관점에서의 분석을 정의하게 되며 시퀀스 다이어그램으로 표현한다.

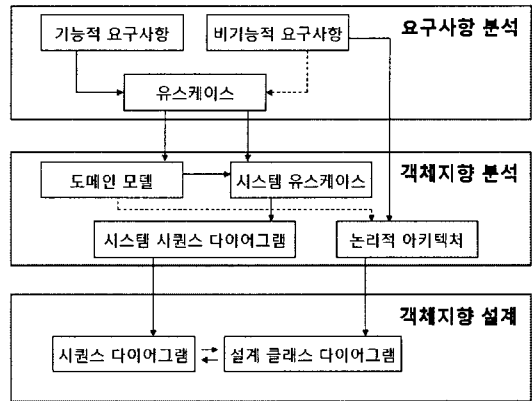


그림 1. RUP의 산출물 관계의 예
Fig. 1. The example of artifact relation from RUP

2.1. 도메인 개념 모델링

웹 콘텐츠 제공자는 콘텐츠를 RSS 포맷에 맞도록 항목들로 정리하여 담아 XML 문서로 만들어 제공한다. 이 XML문서를 RSS 피드(Feed)라고 한다. RSS 구독기는 XML 문서인 RSS 피드를 수집하여 구독자에게 보여주는 역할을 한다. 사용자가 RSS 구독기를 사용하는 환경은 그림 2와 같다.

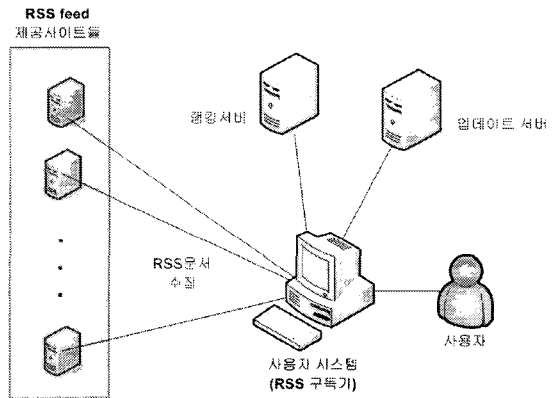


그림 2. RSS 구독기 사용 환경
Fig. 2. Environment of using the RSS Reader

RSS 구독기는 기본적으로 사용자가 웹 상에서 배포하는 RSS 피드 주소를 RSS 구독기에 저장하면 RSS 구독기는 저장된 RSS 피드 주소에서 RSS 피드를 수집하여 저장하고 해당 콘텐츠를 화면에 출력 해주는 것이 요체이다.

RSS 구독기를 설계하기 위한 요구사항은 기능적 요구사항과 비기능적 요구사항으로 나누어 추출하며, 기능적

요구사항들에 기초하여 도메인 개념 모델을 설정하였다. 표1, 2는 RSS 구독기를 설계하기 위한 기능적 요구사항과 비기능적 요구사항을 중요도에 의거하여 상(RSS 구독기의 기능을 하기 위해 꼭 필요한 기능), 중(사용자가 구독기를 사용할 수 있도록 하는 기능), 하(사용자가 구독기를 편리하게 사용할 수 있도록 돕는 기능)로 분류하여 나타낸 것이다. 표 1에서 나타내는 기능적 요구사항은 시스템이 반드시 수행해야 하는 행위를 명기한 것이다.

표 1. 기능적 요구사항
Table. 1 Functional requirements

번호	기능적 요구사항 내용	중요도
1	웹 상에서 배포하는 RSS 피드를 수집하여 저장하고 화면에 출력할 수 있어야 한다.	상
2	사용자가 RSS 피드 주소를 저장, 수정, 삭제할 수 있어야 한다. RSS 피드를 저장할 때 올바른 RSS 피드 주소인지 확인할 수 있어야 한다.	상
3	RSS 피드를 카테고리 방식으로 관리하도록 한다. 카테고리는 사용자 임의로 추가, 삭제, 수정이 가능해야 한다.	중
4	사용자가 원하는 콘텐츠를 찾기 쉽도록 하기 위해 검색을 할 수 있어야 한다.	중
5	사용자가 원하는 내용만을 화면에 출력할 수 있어야 한다.	중
6	외부 애플리케이션 또는 시스템의 북마크 정보에서 RSS 피드를 수집할 수 있어야 한다.	하
7	애플리케이션에 저장되어 있는 데이터를 PDF 포맷으로 변환할 수 있어야 한다.	하
8	RSS 피드 및 콘텐츠들의 랭킹을 제공해 주어야 한다.	하
9	사용자가 RSS 구독기에 저장되어 있는 콘텐츠들의 순위를 매길 수 있어야 한다.	하
10	외부 애플리케이션과 RSS 구독기의 RSS 피드 교환이 용이하도록 RSS 피드를 OPML 포맷으로 저장 또는 OPML 포맷화일에서 RSS 피드를 불러 올 수 있어야 한다.	하
11	기본 브라우저의 웹 주소(URL)를 마우스로 드래그하여 RSS 구독 애플리케이션의 RSS 피드 주소 출력란에 넣으면 RSS 피드 주소가 추가 되어야 한다.	하

표 2는 RSS 구독기의 설계를 위한 비기능적 요구사항을 나타내는 것으로 시스템 자체의 속성을 보인다.

표 2. 비기능적 요구사항
Table. 2 Non-functional requirements

번호	비기능적 요구사항 내용	중요도
1	외부 시스템 및 애플리케이션과 RSS 구독기의 연계가 원활히 이루어져야 한다.	중
2	RSS 피드 추가가 용이하도록 한다.	중
3	무수히 많은 RSS 피드를 수집 할 수 있어야 하고, 이 때 애플리케이션의 성능이 저하되지 않도록 해야 한다.	상
4	향후에 RSS 이외의 포맷들도 지원할 수 있도록 해야 한다.	중
5	RSS 구독기의 기능 추가가 용이 하도록 해야 한다.	중
6	저장되어 있는 데이터를 PDF 포맷으로 변환할 때 2초를 넘지 않도록 해야 한다.	하
7	많은 사용자가 사용할 수 있도록 하기 위해 윈도우, 리눅스, 맥 등 플랫폼에 종속적이지 않아야 한다.	중
8	사용자가 RSS 피드의 수집과 같이 RSS 구독기에 포함된 기능을 실행시킬 수 있어야 하고, RSS 피드를 제공하는 사이트 및 RSS 피드의 정보들을 확인할 수 있어야 한다.	상

도메인 개념 모델은 도메인 안에 존재하는 개념 또는 클래스와 그들 간의 관계를 시각적으로 표현한 것이다 [12][13]. 본 논문에서는 요구사항 분석 모델[14]에서 제시된 모델을 확장, 개선하였으며, 그 결과는 아래의 그림과 같다.

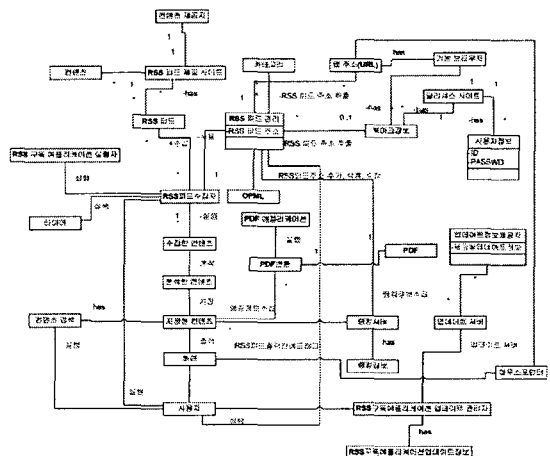


그림 3. RSS 구독기의 도메인 개념 모델 설정
Fig. 3. The domain concept model configuration of the RSS Reader

그림 3은 RSS 피드 제공 사이트와 콘텐츠 제공자들의 사용자 시스템의 외부 도메인과 RSS 피드 수집자, 수집한 콘텐츠 등과 같은 내부 도메인을 함께 표현하며 RSS 구독기와 이것과 관련된 도메인 개념을 하나의 모델로 나타내었고, 이렇게 함으로써 RSS 구독기의 내, 외부 도메인들의 연결 설정 등을 시각적으로 확인하였다. 그림3에서 나타내는 도메인은 요구사항 분석 모델의 기능적 요구사항에서 명사구를 식별하여 도메인을 추출하였고 추출한 도메인들의 활동을 식별하여 관련 도메인을 추출하는 방법을 이용하였다. 이렇게 명사구를 먼저 식별하여 도메인을 찾는 방법은 명사구가 여러 의미를 내포한 경우에 다소 혼동이 되어질 수 있지만, 도메인을 찾는 데 있어서 보다 많은 편의성이 제공되는 장점이 있다[15].

2.2. 유스케이스 및 동작 모델링

앞 절에서 분석한 RSS 구독기의 도메인 모델로부터 추출한 시스템 유스케이스는 그림 4와 같다.

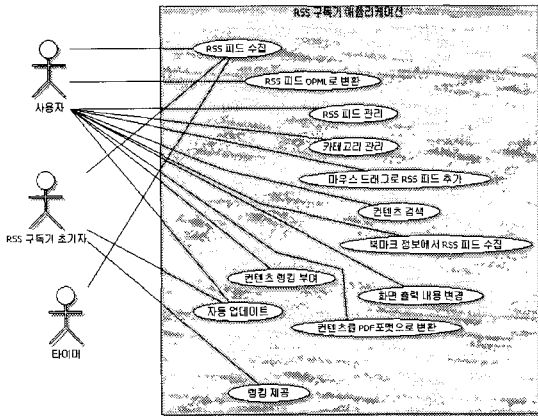


그림 4. RSS 구독기의 시스템 유스케이스
Fig. 4. The system use case of the RSS Reader

각 유스케이스들은 액터들에 의해 촉발되며, 주요 유스케이스를 살펴보면 'RSS 피드 수집' 유스케이스는 콘텐츠를 제공하는 외부 웹 사이트에서 RSS 피드를 수집하고 수집한 콘텐츠를 저장하는 유스케이스이다. 이 유스케이스는 사용자와 RSS 구독기 초기자, 타이머 등의 액터와 연관되어 있다. 즉, RSS 구독기가 실행되어 RSS 구독기 초기자 액터가 실행되거나, RSS 구독기에 RSS 피드 수집 타이머를 설정해 놓거나, 또는 사용자가 RSS 피드 수집을 실행하면 RSS 구독기는 RSS 피드를 수집하게 된다.

시스템 시퀀스 모델은 유스케이스의 특정 시나리오를 바탕으로, 외부의 액터가 발행시키는 이벤트, 이벤트의 순서, 시스템 간의 이벤트를 보여주는 다이어그램에 의해 표현된다[10]. 그림 4의 시스템 유스케이스에 기초하여 설계한 RSS 구독기의 시스템 시퀀스 모델을 그림 5에서 보인다.

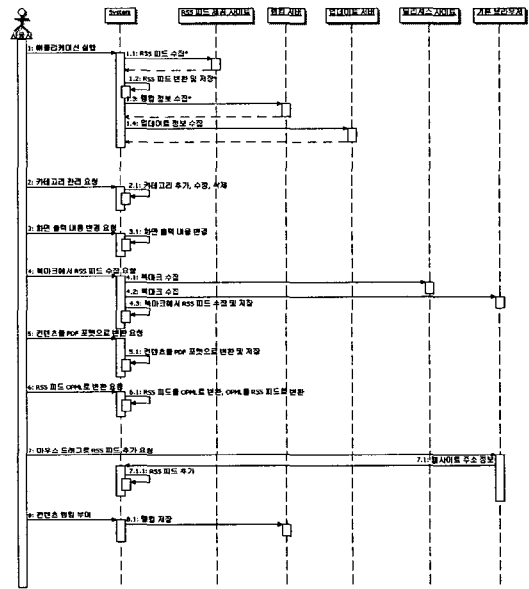


그림 5. RSS 구독기의 시스템 시퀀스 모델
Fig. 5. The system sequence model of the RSS Reader

2.3. 아키텍처 설계

논리적 아키텍처는 소프트웨어의 클래스들을 패키지, 서브시스템, 계층과 같은 큰 단위로 구성한 것이다. RSS 구독기의 논리적 아키텍처를 설계한 결과는 그림 6과 같다.

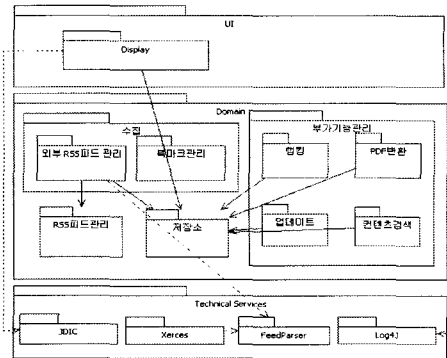


그림 6. RSS 구독기의 논리적 아키텍처
Fig. 6 The logical architecture of the RSS Reader

위의 그림 6에서 나타내는 UI계층은 GUI윈도우 등 사용자와 인터페이스를 할 수 있도록 하는 계층으로 이곳에는 Display 패키지를 두어 RSS 구독기의 화면출력과 사용자 인터페이스에 관련된 모든 클래스들로 구성한다.

Domain 계층은 애플리케이션의 요구사항을 만족시켜 주기 위한 도메인 내의 개념을 나타내는 계층이다. 수집, 랭킹, RSS피드관리, 부가기능관리 등 4개의 패키지는 이 계층에 만족하는 개념을 내재하므로 Domain계층에 포함한다. Technical Service계층은 기술적 서비스를 지원하기 위한 계층으로 JDIC[16], Xerces[17], FeedParser[18], Log4J[19]등의 서브시스템을 포함한다.

Ⅲ. RSS 구독기 설계

3.1. 클래스와 패키지

설계 클래스 다이어그램은 설계 관점으로 시스템 내부의 정적인 객체를 표현한 것이다. 앞에서 RSS 구독기를 분석한 산출물들을 토대로 추출하고 설계한 클래스들을 보이면 그림 7과 같다.

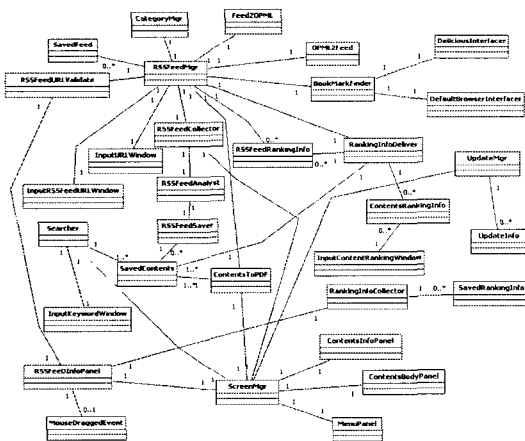


그림 7. RSS 구독기의 클래스
Fig. 7. The classes of the RSS Reader

그림 7에 나타나는 RSSFeedCollector 클래스는 RSS피드를 수집하는 기능을 수행하는 클래스이다. 이 클래스는 수집한 RSS피드를 분석하는 기능을 수행하는 RSSFeedAnalyst 클래스와 RSS피드를 관리하는 기능을 수행하는 RSSFeedMgr 클래스와 각각 1:1 연관을 맺고 있다. 본 논문에서는 그림 7에 나타나는 클래스들에 기초하여 기능적인 측면에서

Repository, Collection, RSSFeedMgt, AddFuncMgt, Display 패키지로 나누어 설계하였다.

3.1.1. Repository 패키지

Repository 패키지는 다른 패키지에서 사용하거나 RSS 구독기에서 전반적으로 사용하는 정보 클래스들을 모아 관리하는 패키지이다. 이렇게 정보 클래스들을 모아 패키징 하면 시스템의 응집력이 낮아지고, 결합도가 높아지고, 패키지 설계 시 복잡도가 높아질 수 있으나, 시스템의 설계를 기반으로 하여 구현할 때 정보 클래스들의 관리가 편리한 장점을 가지고 있다. 그림 8은 Repository 패키지를 나타낸 다이어그램이다.

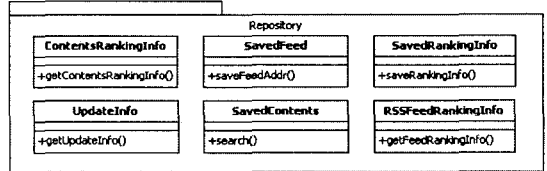


그림 8. Repository 패키지 다이어그램
Fig. 8. The Repository package diagram

Repository 패키지의 클래스들은 Repository이외의 패키지에서 사용하는 정보에 대한 CRUD(Create, Retrieve, Update, Delete)를 이행한다.

3.1.2. Collection 패키지

ExternalRSSFeedMgt 패키지와 BookmarkMgt 패키지를 포함하고 있는 Collection패키지는 RSS 피드를 수집하는 기능을 수행하는 패키지이다. Collection 패키지를 나타낸 다이어그램은 그림 9와 같다.

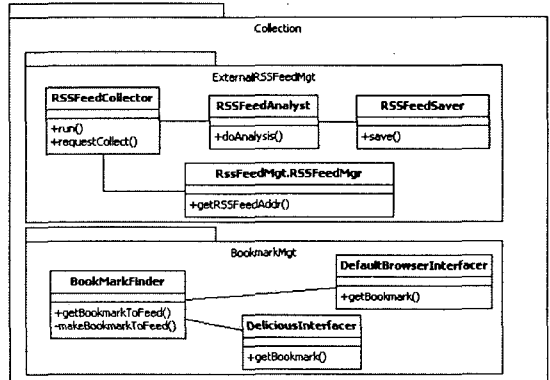


그림 9. Collection 패키지 다이어그램
Fig. 9. The Collection package diagram

ExternalRssFeedMgt 패키지는 RSS 제공 사이트에서 RSS 피드를 수집하는 기능을 하며, 수집한 RSS 피드를 분석, 저장하는 클래스들을 포함하고 있다. 이 기능을 수행하는 절차는 그림 10과 같다.

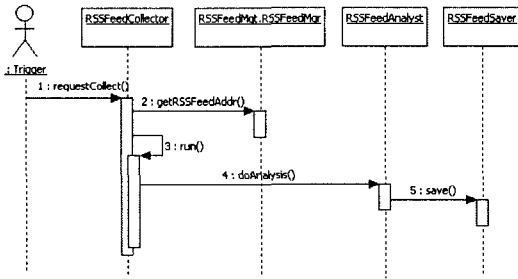


그림 10. ExternalRssFeedMgt 패키지의 시퀀스 다이어그램
Fig. 10. The sequence diagram of the ExternalRssFeedMgt package

그림 10은 Trigger가 RSS 피드의 수집을 요청하면 요청한 RSS 피드의 주소에서 RSS 피드를 수집하고, 이것을 해석한 후 시스템에 저장하는 절차를 나타낸다.

북마크에서 RSS 피드를 수집하는 BookmarkMgt 패키지는 외부에 있는 딜리셔스 사이트와 사용자 시스템의 기본 브라우저를 연결할 수 있는 클래스와 이들에게서 RSS 피드를 수집하는 클래스로 이루어져 있다. 이들의 순차적인 처리절차는 그림 11과 같은 순으로 진행된다.

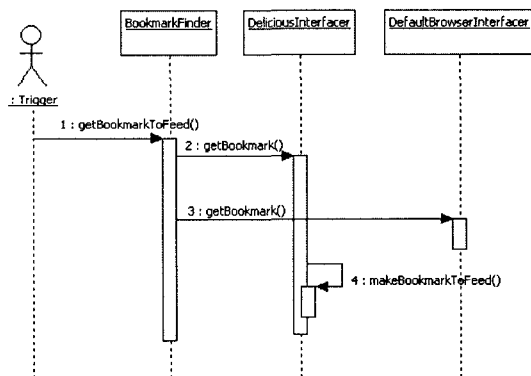


그림 11. BookmarkMgt 패키지의 시퀀스 다이어그램
Fig. 11. The sequence diagram of the BookmarkMgt package

그림 11에서 Trigger가 북마크의 RSS 피드를 요청하면 딜리셔스와 시스템의 기본 브라우저에서 북마크 정보를

찾고 이 정보에서 RSS 피드를 찾아 반환하는 절차를 확인할 수 있다.

3.1.3. RSSFeedMgt 패키지

RSS 피드는 OPML(Outline Processor Markup Language) 포맷 S파일로 관리할 수 있는데, OPML은 XML기반의 파일 포맷으로 각기 다른 환경과 운영체제의 애플리케이션 실행 사이에서 개요 구조 정보를 교환하는 포맷이다[20]. RSSFeedMgt 패키지는 OPML을 이용한 RSS 피드 관리 등과 같이 RSS 피드와 연관된 기능들을 수행하는 클래스들을 포함하는 패키지이다. RSSFeedMgt 패키지를 나타낸 다이어그램은 그림 12와 같다.

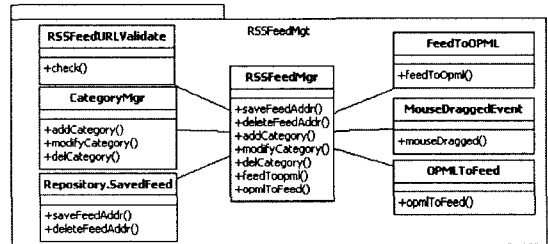


그림 12. RSSFeedMgt 패키지 다이어그램
Fig. 12 The RSSFeedMgt package diagram

RSSFeedMgt 패키지의 기능들을 수행하는 절차는 그림 13과 같다.

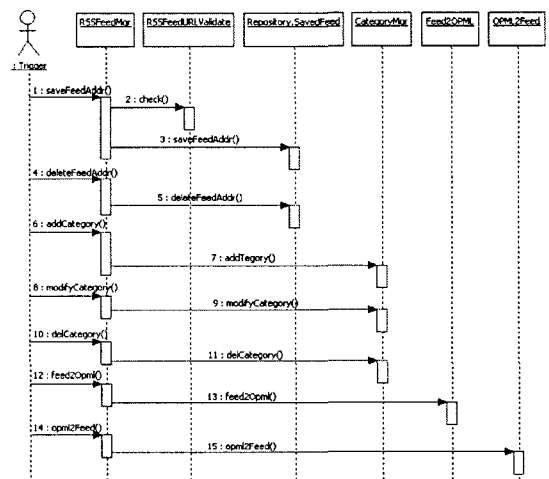


그림 13. RssFeedMgt 패키지의 시퀀스 다이어그램
Fig. 13. The sequence diagram of the RssFeedMgt package

그림 13은 RSS 피드 주소를 저장, 삭제하는 절차와 카테고리 추가, 수정, 삭제 하는 절차, 그리고 RSS 피드를 OPML로 저장하거나, OPML파일에서 RSS 피드를 불러오는 절차를 나타낸다.

3.1.4. AddFuncMgt 패키지

RSS 구독기의 부가적인 기능들을 포함하는 AddFuncMgt 패키지는 Ranking, SearchContents, Update, PDFMgt 패키지를 포함한다. AddFuncMgt 패키지를 나타낸 다이어그램은 그림 14와 같다.

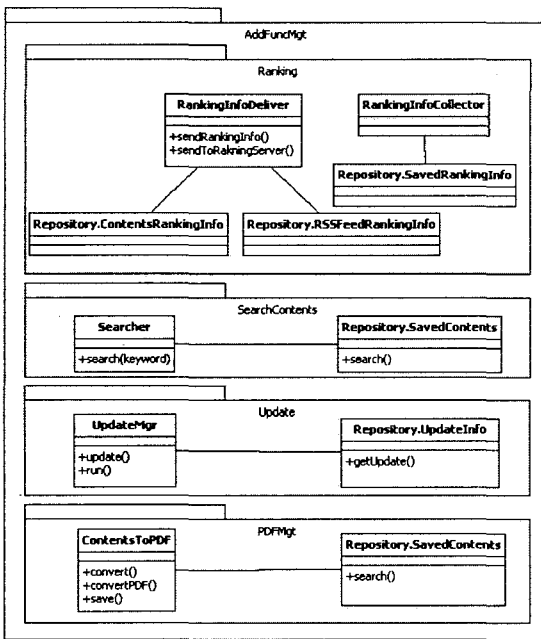


그림 14. AddFuncMgt 패키지 다이어그램
Fig. 14. The AddFuncMgt package diagram

AddFuncMgt 패키지에 포함된 Ranking 패키지는 RSS 피드와 콘텐츠들의 랭킹정보를 화면에 출력하거나 수집하는 기능을 수행하는 클래스들을 포함한다. Ranking 패키지에 포함되어 있는 클래스들이 수행하는 기능의 순차적인 처리 절차는 그림 15와 같은 순으로 진행된다.

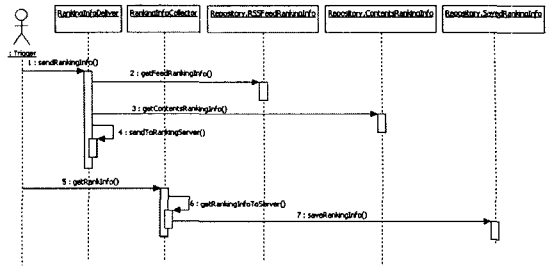


그림 15. Ranking 패키지의 시퀀스 다이어그램
Fig. 15. the sequence diagram of the Ranking package

그림 15에서 나타내는 랭킹정보를 서버로 전송하는 절차에서 Trigger는 랭킹정보를 서버로 전송할 것을 요청하고, RSS 피드와 콘텐츠의 랭킹정보를 서버로 전송한다. 또한, 랭킹정보를 서버에서 수집하기 위하여 Trigger가 시스템에게 랭킹정보를 수집할 것을 요청하면, 랭킹정보를 수집하고, 저장한다.

SearchContents 패키지는 저장되어 있는 콘텐츠를 입력받은 키워드로 검색하는 기능을 수행하는 클래스를 포함하고 있으며, 이를 위한 절차는 그림 16과 같다.

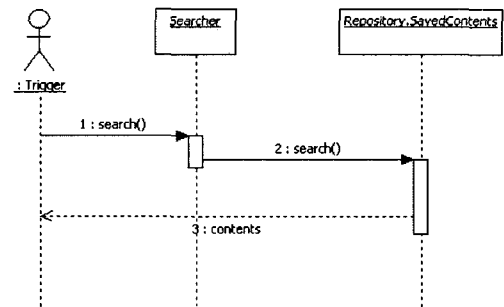


그림 16. SearchContents 패키지의 시퀀스 다이어그램
Fig. 16. The sequence diagram of the SearchContents package

그림 16에서 Trigger가 키워드를 입력하여 검색을 요청하면 검색을 하고 검색 결과를 반환하는 절차를 확인할 수 있다.

AddFuncMgt 패키지가 포함하고 있는 패키지들 중 Update 패키지는 저장되어 있는 콘텐츠에서 입력받은 키워드로 검색하는 기능을 수행하는 클래스를 포함하고 있으며, 이를 위한 절차는 그림 17과 같다.

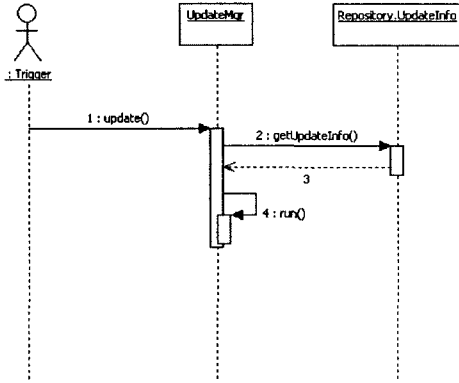


그림 17. Update 패키지의 시퀀스 다이어그램
Fig. 17. The sequence diagram of the Update package

그림 17은 Trigger가 자동 업데이트를 요청하고, 시스템은 업데이트의 판별을 위해 저장되어 있는 업데이트 정보를 확인하고, 업데이트 서버의 정보와 비교하여 실행하는 자동 업데이트 절차를 나타낸다.

PDFMgt 패키지는 저장된 콘텐츠를 PDF 포맷 파일로 변환하는 기능을 수행하는 클래스를 포함하고 있으며, 이 기능을 수행하는 절차는 그림 18과 같다.

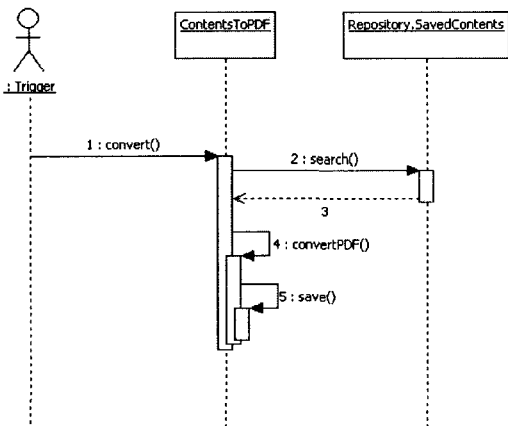


그림 18. PDFMgt 패키지의 시퀀스 다이어그램
Fig. 18. The sequence diagram of the PDFMgt package

그림 18은 저장되어 있는 콘텐츠를 PDF로 저장하기 위한 절차를 나타낸다. Trigger가 키워드를 포함한 콘텐츠를 PDF 포맷으로 변환할 것을 요청하면 시스템은 키워드로 콘텐츠를 검색, 확인한 결과를 PDF 파일로 변환, 저장한다.

3.1.5 Display 패키지

Display 패키지는 RSS구독기의 화면 출력에 관련된 기능을 하는 클래스들을 포함하고 있다. Display 패키지를 나타낸 다이어그램은 그림 19와 같다.

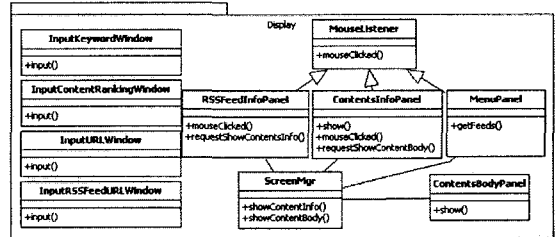


그림 19. Display 패키지 다이어그램
Fig. 19. The Display package diagram

그림 19에 나타난 Display 패키지가 포함하는 클래스에서 콘텐츠 및 콘텐츠 정보를 화면에 출력하기 위한 절차는 그림 20과 같은 순으로 진행된다.

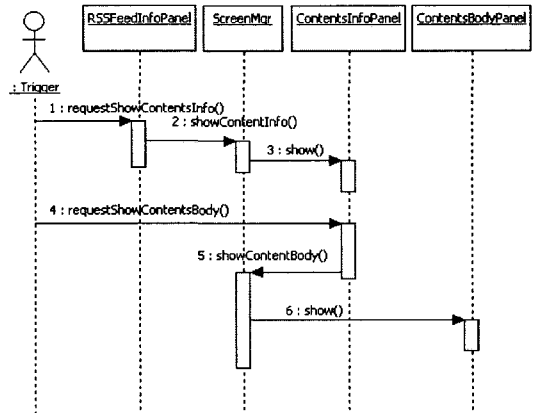


그림 20. Display 패키지의 시퀀스 다이어그램
Fig. 20. The sequence diagram of the Display package

그림 20에서 확인할 수 있는 화면 출력 절차는 Trigger가 콘텐츠를 정보를 화면에 출력하기 위한 요청을 하면 요청한 콘텐츠를 검색하여 화면에 출력하고, Trigger가 콘텐츠의 본문 정보의 화면 출력을 요청하면 이것을 검색하여 화면에 출력하는 절차이다.

3.2. 프리젠테이션 설계

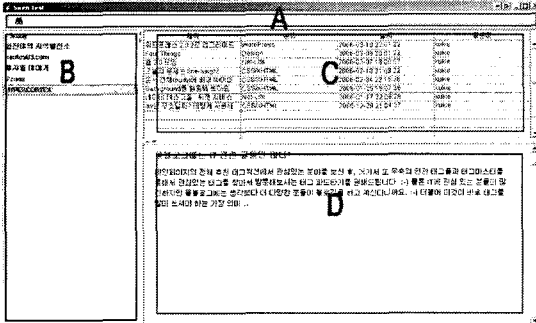


그림 21. RSS 구독기의 UI 설계
Fig. 21. The UI design of the RSS Reader

RSS 구독기의 프리젠테이션은 비기능적 요구사항 8에 의거하여, 사용자가 쉽게 사용할 수 있도록 기존의 타 애플리케이션과 비슷한 UI가 되도록 설계하였다. (그림 21) 그림 21의 RSS 구독기는 3.1.2 절의 Collection 패키지를 이용하여 RSS 피드를 수집한다. 수집된 정보들은 Repository 패키지에 의해 관리되며, 이 패키지에서 추출된 정보가 출력된 화면이 그림 21이다. 표 1 기능적 요구사항에 보인 RSS 피드 수집, 업데이트 기능 등을 수행하기 위하여 A 부분을 설계하였고, 구독중인 사이트를 보이기 위하여 B 부분을 설계하였다. B 부분에서 수행되는 RSS 피드 관리 기능들은 3.1.3 절의 RSSFeedMgt 패키지에서 수행한다. C 부분은 3.1.4 절에서 설계한 AddFuncMgt 패키지의 Ranking 패키지와 Repository 패키지를 이용하여 랭킹정보와 콘텐츠 정보들이 화면에 출력되도록 하였고, C 부분의 콘텐츠 정보와 관련된 콘텐츠의 본문 내용을 Repository 패키지를 이용하여 D 부분에 출력되도록 하였다. 그림 21의 B, C, D 부분은 모두 연계되어 있다. 즉, B 부분의 RSS 피드를 선택하면, 선택된 RSS 피드의 콘텐츠 정보가 C 부분에 출력되고, C 부분의 콘텐츠를 선택하면, D 부분에 콘텐츠의 본문이 출력된다.

그림 22는 RSS 구독기(RSS Client)가 콘텐츠 제공 사이트(RSS Server)에 접속해서 RSS 피드를 읽어오기까지의 처리 절차를 나타낸 것이다.

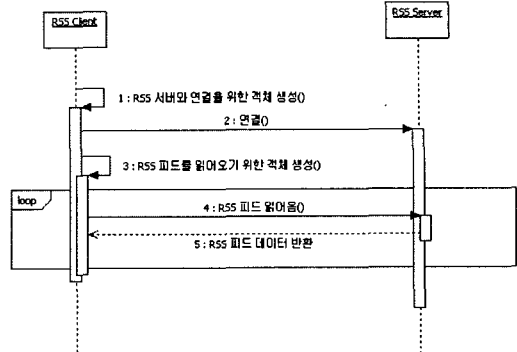


그림 22. RSS 클라이언트가 RSS 서버의 RSS 피드를 수집하는 절차
Fig. 22. The process that the RSS client collect the RSS Feed from the RSS server

그림 23은 그림 22에서 보인 RSS Client가 RSS Server의 RSS 피드를 수집하는 처리의 절차를 자바 프로그래밍 언어로 구현한 소스를 나타낸다.

```

01: public RSSFeedCollect extends Thread
02: { .....
03:     public void run()
04:     { .....
05:         //RSS 클라이언트와 RSS 서버의 연결설정 부분
06:         String urlStr = "http://example.com/rss";
07:         URL url = new URL( urlStr );
08:         URLConnection connect = url.openConnection();
09:         connect.connect();
10:         //RSS 서버의 RSS 피드를 읽는 부분
11:         InputStream in = connect.getInputStream();
12:         byte[] buf = new byte[1024];
13:         int len;
14:         while((len = in.read(buf)) >= 0)
15:         {
16:             //읽어온 데이터를 해석 저장한다
17:             .....
18:         }
19:     }
20: }
    
```

그림 23. RSSFeedCollect 클래스의 RSS 서버 연결 설정 및 RSS 피드를 읽는 소스

Fig 23. The connected configuration of the RSS server and the source which needs to read the RSS Feed in RSSFeedCollect class

먼저, RSS 클라이언트와 RSS 서버의 연결 설정을 위하여 RSS 피드 주소를 가진 URL 객체를 만들고(09라인) RSS Server와 연결을 설정하기 위해 URLConnection 객체를 생성한다.(08라인) 이 객체의 connect() 메소드를 이용하여 연결 설정을 완료한다.(09라인) 다음으로, 연결된 RSS 서버의 RSS 피드를 읽는 부분은 RSS 서버와 연결한 URLConnection 오브젝트를 이용하여 InputStream 오브젝트를 생성하고(11라인), 이 오브젝트의 read() 메소드를 이용하여 설정된 버퍼(12라인)만큼 반복하여 RSS 피드를 읽어오도록 한다.(14라인) 모든 RSS 피드를 읽어오면 전송을 완료한다.

IV. 결 론

본 논문은 RSS 구독기 애플리케이션을 논리적인 관점으로 3개의 계층으로 나누고, 이것을 기반으로 하여 패키징하는 방법으로 설계하여 시스템 결합도와 의존관계를 감소시키고, 응집도를 향상시켰으며, 재사용 가능성과 명료성을 높였다. 또한, 논리적인 부분들로 나누었기 때문에 팀 단위의 개발이 수월해질 수 있는 장점을 가지도록 하였다. 이렇게 설계한 애플리케이션은 사용자의 시스템에 설치하여 사용하는 데스크탑 기반의 응용프로그램이므로 RSS 구독기를 설치한 시스템 이외의 시스템에서는 사용할 수 없다는 단점이 있으나 웹 기반의 RSS 구독기보다 사용자가 구독한 자료를 개인이 관리할 수 있다는 장점이 있다.

타 RSS 구독기와 비교하여 기본 브라우저 및 북마크 관리 사이트인 딜리셔스 사이트와의 연동으로 사용자가 RSS 구독기를 사용함에 있어 가장 번거러움을 느낄 수 있는 RSS 피드의 추가를 편리하게 하였다. 그리고 랭킹 기능을 추가하여 사용자들이 선호하는 RSS 피드 제공 사이트 및 콘텐츠를 확인할 수 있도록 하였고, OPML과 PDF 포맷을 지원함으로써 사용자가 관리하는 정보의 고립을 막도록 설계하였다. 하지만, 이러한 기능들 때문에 사용자의 시스템 자원을 많이 사용할 수 있고 이러한 이유로 실행 속도가 감소될 수 있다. 그리고 랭킹 및 자동 업데이트 기능을 제공하기 위해 랭킹 또는 업데이트 서버 등의 물리적인 장치가 필수적이어야 하고, 랭킹 기능에서 사용자의 정확한 랭킹 정보 입력이 이루어 지지 않거나, 참여 횟수가 적으면 신뢰도가 떨어지는 랭킹정보를 제공하게

된다는 단점을 가지고 있다.

본 논문에서는 RSS 구독기의 구현을 위하여 RUP 객체지향 설계 방법론에 의거한 체계적인 소프트웨어 설계를 제시하였다. 요구사항 분석 모델 [14]에서 제시된 모델을 확장, 개선하여 도메인 개념 모델과 유스케이스 모델, 그리고 유스케이스의 액터와 시스템간의 이벤트를 식별할 수 있는 시스템 시퀀스 모델을 제시하였고, 이들을 바탕으로 RSS 구독기의 논리적 아키텍처를 설계하여 클래스와 패키지의 설계에 발판이 되도록 하였다. 그리고, 클래스를 여러 개의 패키지로 나누고 각 패키지의 클래스의 연관관계를 보였다. 또한 UML의 시퀀스 다이어그램을 이용하여 이들의 오브젝트들의 수행 절차들을 나타내어 각 오브젝트들의 시간에 따른 흐름을 보였다. 마지막으로 RSS 구독기의 사용자 UI의 설계화면을 보이고, RSS 구독기가 RSS 피드를 수집하는 과정을 자바 프로그램 언어로 나타내었다. 이와 같이 아직 초기 단계에 있는 RSS 구독기의 체계적인 설계 방법을 제시함으로써, 향후에 이를 기반으로 하여 손쉽고 다양한 기능을 가진 RSS 구독기의 제작이 활발히 이루어지기를 기대한다.

참고문헌

- [1] The RSS 1.0 specification, <http://web.resource.org/rss/1.0/spec>
- [2] The RSS 2.0 specification, <http://blogs.law.harvard.edu/tech/rss>
- [3] Ben Hammersley, *Developing Feeds with RSS and ATOM*, O'Reilly Media, 2005
- [4] 연모, <http://yeonmo.theple.com>
- [5] 샤프리더, <http://www.sharpreader.net>
- [6] 엑스파이더, <http://www.xpyder.co.kr>
- [7] 한RSS, <http://hanrss.com>
- [8] 블로그라인스 <http://www.bloglines.com>
- [9] rojo, <http://rojo.com>
- [10] Larman Craig, *Apply uml and patterns : An introduction to object-oriented analysis and design and iterative development*, 3rd Edition, Prentice Hall, 2005
- [11] James Rumbaugh, Ivar Jacobson, Grady Booch, *Unified Modeling Language Reference Manual*, 2nd Edition, Addison-Wesley, 2004

- [12] Martin James, James J. Odell, *Object-Oriented Methods: A Foundation*, 2nd Edition, Prentice-Hall, 1995
- [13] Fowler, *Analysis Patterns: Reusable Object Models*, Addison-Wesley, 1996
- [14] 이동규, 김윤호, “RSS 리더기를 위한 요구사항 분석 모델”, *한국해양정보통신학회 2006 춘계종합학술대회*, Vol.10 No.1, pp569-572, 2006
- [15] Abbott R. J., “Program Design by Informal English Description”, *Communications of the ACM* Vol.26 No.11, p882-894, 1983
- [16] JDIC, <http://jdic.dev.java.net/>
- [17] Xerces, <http://xerces.apache.org/>
- [18] FeedParser, <http://jakarta.apache.org/commons/sandbox/feedparser/>
- [19] Log4J, <http://logging.apache.org/log4j>
- [20] Outline Processor Markup Language, <http://www.opml.org/>

저자소개



이 동 규(Dong-kyu Lee)

2005. 2 안동대학교 컴퓨터공학과 공학사
2005. 3 - 현재 안동대학교 컴퓨터공학과 석사과정

※ 관심분야 : 객체지향 분석/설계/프로그래밍



김 윤 호(Yun-Ho Kim)

1983. 2. 경북대학교 전자공학과 공학사
1993. 2. 경북대학교 대학원 컴퓨터공학과 공학석사

1997. 2. 경북대학교 대학원 컴퓨터공학과 공학박사
1997. 8. - 현재 안동대학교 전자정보산업학부 부교수
※ 관심분야 : 인터넷 컴퓨팅, 객체지향 분석/설계/프로그래밍, 분산객체시스템, 병렬처리