INRIA	**RSS to ATOM** **ATOM to RSS**	**Author** **SROUR Youssef** **Srour.youssef@gmail.com**
		July Release 0.1

RSS

RSS is a format for syndicating news and the content of news-like sites, including major news sites like Wired, news-oriented community sites like Slashdot, and personal web logs. But it's not just for news. Pretty much anything that can be broken down into discrete items can be syndicated via RSS: the "recent changes" page of a *wiki*, a change log of CVS chickens, even the revision history of a book. Once information about each item is in RSS format, an RSS-aware program can check the feed for changes and react to the changes in an appropriate way.

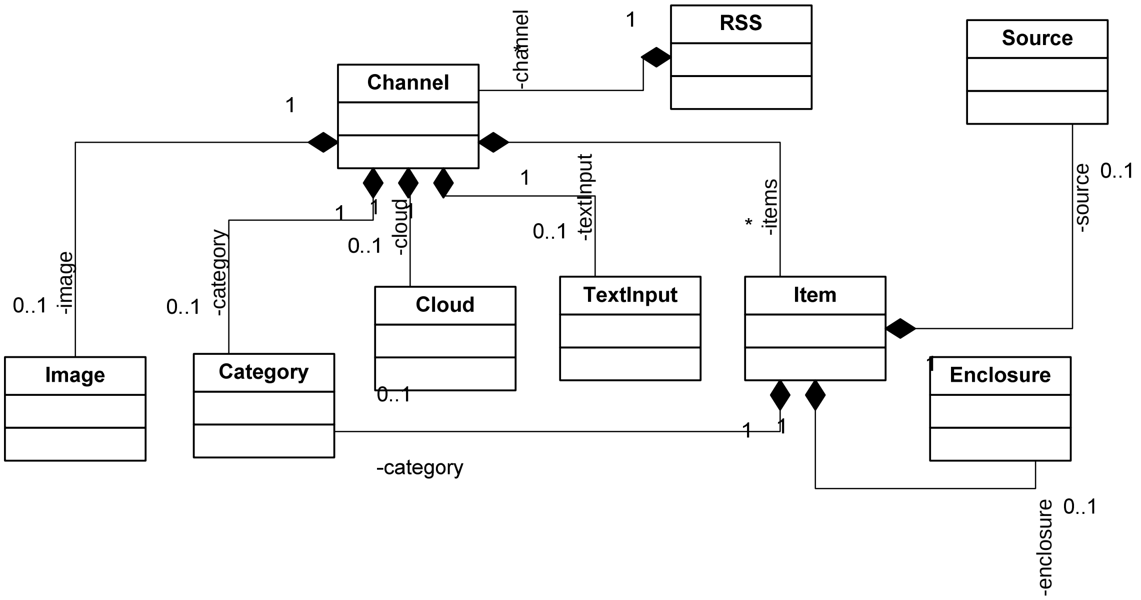
RSS-aware programs called news aggregators are popular in the web logging community. Many web logs make content available in RSS. A news aggregator can help you keep up with all your favorite web logs by checking their RSS feeds and displaying new items from each of them.

See http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html

I. Meta model of RSS in KM3

```
-- @name          RSS-2.0
-- @version       2.1
-- @domains
-- @authors       David Touzet (david.touzet@univ-nantes.fr)
-- @date          2005/11/01
-- @Modified by Youssef Srour (srour.youssef@gmail.com)  24/06/2006
-- @description   This metamodel describes RSS.
-- @see http://blogs.law.harvard.edu/tech/rss
-- @see http://en.wikipedia.org/wiki/RSS_(protocol)

package RSS {

--Begin class RSS
      class RSS {
            attribute version : String;
            reference channel container : Channel oppositeOf rss;
      }--End class RSS

--Begin class Channel
      class Channel {
            attribute title : String;
            attribute link[0-1] : String;
            attribute description : String;
            attribute language[0-1] : String;
            attribute copyright[0-1] : String;
            attribute managingEditor[0-1] : String;
            attribute webmaster[0-1] : String;
            attribute generator[0-1] : String;
            attribute docs[0-1] : String;
            attribute ttl[0-1] : Integer;
            attribute rating[0-1] : String;
            attribute skipHours[*] : Integer;
            attribute pubDate[0-1] : String;
            attribute skipDays[1-*] : String;
            attribute lastBuildDate[0-1] : String;

            reference rss : RSS oppositeOf channel;
            reference image[0-1] container : Image oppositeOf channel;
            reference textInput[0-1] container : TextInput oppositeOf
channel;
            reference cloud[0-1] container : Cloud oppositeOf channel;
            reference category[0-1]  container : Category oppositeOf
channel;
            reference items[*] container : Item oppositeOf channel;
      }--End class Channel

--Begin class Item
      class Item {
            attribute title : String;
            attribute link : String;
            attribute description : String;
            attribute pubDate[0-1] : String;
            attribute author[0-1] : String;
            attribute comments[0-1] : String;
            attribute guid[0-1] : String;

            reference source[0-1] container : Source;
            reference enclosure[0-1] container : Enclosure;
            reference category[0-1] container : Category oppositeOf items;
            reference channel : Channel oppositeOf items;
      }--End class Item

--Begin class Image
      class Image {
            attribute url : String;
            attribute title : String;
            attribute link : String;
            attribute description[0-1] : String;
            attribute width[0-1] : Integer;
            attribute height[0-1] : Integer;

            reference channel : Channel oppositeOf image;
      }--End class Image

--Begin class TextInput
      class TextInput {
            attribute title : String;
            attribute description : String;
            attribute name : String;
            attribute link : String;

            reference channel : Channel oppositeOf textInput;
      }--End class TextInput

--Begin class Cloud
      class Cloud {
            attribute domain : String;
            attribute port : Integer;
            attribute path : String;
            attribute registerProcedure : String;
            attribute protocol : String;

            reference channel : Channel oppositeOf cloud;
      }--End class Cloud

--Begin class Category
      class Category {
            attribute domain : String;
            attribute value : String;

            reference channel : Channel oppositeOf category;
            reference items : Item oppositeOf category;
      }--End class Category

--Begin class Enclosure
      class Enclosure {
            attribute url : String;
            attribute lenght : Integer;
            attribute type : String;
      }--End class Enclosure

--Begin class Source
      class Source {
            attribute url : String;
            attribute value : String;
      }--End class Source
}

package PrimitiveTypes {
      datatype String;
      datatype Integer;
}
```

II. Graphical Meta model of RSS in UML



ATOM

Atom is an XML-based file format intended to allow lists of information, known as "feeds", to be synchronised between publishers and consumers. Feeds are composed of a number of items, known as "entries", each with an extensible set of attached metadata. For example, each entry has a title.
The primary use case that Atom addresses is for syndicating Web content such as Web logs and news headlines to other Web sites and directly to consumers. However, nothing precludes it from being used for other purposes and types of content.
Details of communication protocols between software agents using Atom are to be found in the Atom API.

See http://www.mnot.net/drafts/draft-nottingham-atom-format-02.html

I. Meta model of ATOM in KM3

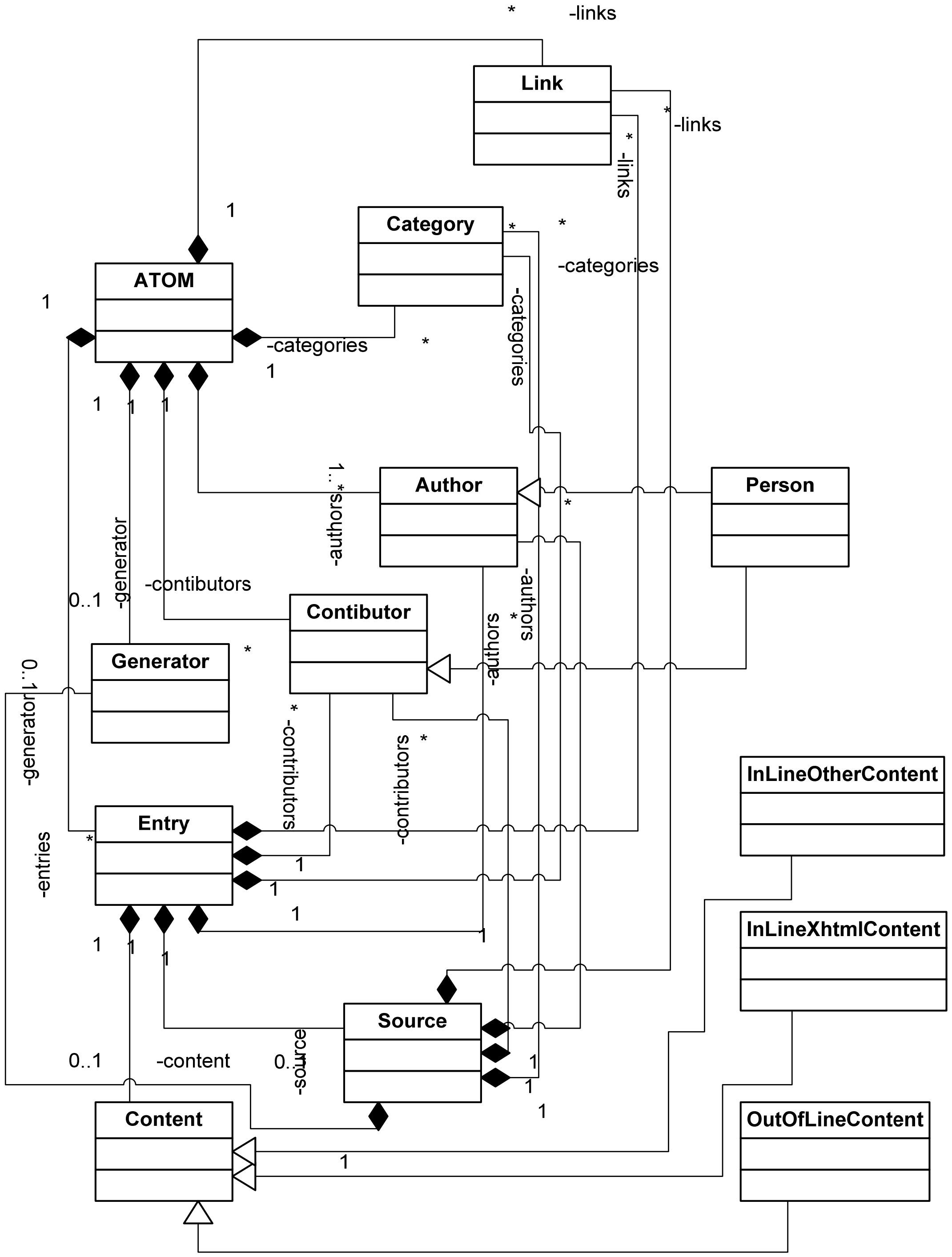
```
-- @name          ATOM 1.0
-- @version       1.2
-- @domains
-- @authors       David Touzet (david.touzet@univ-nantes.fr)
-- @date          2005/11/01
-- @Modified by Youssef Srour (srour.youssef@gmail.com)  24/06/2006
-- @description   This metamodel describes ATOM.
-- @see http://www.atomenabled.org/developers/syndication/
-- @see http://en.wikipedia.org/wiki/Atom_%28standard%29

package ATOM {

-- Begin Class ATOM
      class ATOM {
            attribute title : String;
            attribute id : String;
            attribute subtitle[0-1] : String;
            attribute rights[0-1] : String;
            attribute icon[0-1] : String;
            attribute logo[0-1] : String;
            attribute lastUpdate  : String;

            reference links[1-*] container : Link oppositeOf atom;
            reference generator[0-1] container : Generator;
            reference categories[0-*] container : Category oppositeOf atom;
            reference authors[1-*] container : Author oppositeOf atom;
            reference contributors[*] container : Contributor;
            reference entrie[*] container : Entry oppositeOf atom ;
      }-- End Class ATOM
```

```
-- Begin Class Entry
      class Entry {
            attribute title : String;
            attribute id : String;
            attribute rights[0-1] : String;
            attribute summary[0-1] : String;
            attribute published[0-1] : String;
            attribute lastUpdate : String;

            reference links[1-*] container : Link oppositeOf entry;
            reference source[0-1] container : Source;
            reference content[0-1] container : Content;
            reference categories[*] container : Category oppositeOf entry;
            reference authors[1-*] container : Author ;
            reference contributors[*] container : Contributor;
            reference atom : ATOM oppositeOf entrie;
      }-- End Class Entry
-- Begin Class Source
      class Source {
            attribute id[0-1] : String;
            attribute icon[0-1] : String;
            attribute logo[0-1] : String;
            attribute rights[0-1] : String;
            attribute title[0-1] : String;
            attribute subtitle[0-1] : String;
            attribute lastUpdate[0-1] : String;

            reference links[*] container : Link;
            reference generator[0-1] container : Generator;
            reference contributors[*] container : Contributor;
            reference categories[*] container : Category;
            reference author[0-1] container : Author;
      }-- End Class Source

-- Begin Class Content
      abstract class Content {
            attribute type[0-1] : String;
            attribute mode[0-1] : String;
            attribute text[0-1] : String;
      }-- End Class Content

-- Begin Class InLineTextContent
      class InLineTextContent extends Content {
      }-- End Class InLineTextContent

-- Begin Class InLineXHTMLContent
      class InLineXHTMLContent extends Content {
      }-- End Class InLineXHTMLContent

-- Begin Class InLineOtherContent
```

```
    class InLineOtherContent extends Content {
    }-- End Class InLineOtherContent

-- Begin Class OutOfLineContent
    class OutOfLineContent extends Content {
        attribute src : String;
    }-- End Class OutOfLineContent

-- Begin Class Generator
    class Generator {
        attribute uri[0-1] : String;
        attribute version[0-1] : String;
        attribute name : String;
    }-- End Class Generator

-- Begin Class Category
    class Category {
        attribute term : String;
        attribute scheme[0-1] : String;
        attribute label[0-1] : String;
        reference atom : ATOM oppositeOf categories;
        reference entry : Entry oppositeOf categories;
    }-- End Class Category

-- Begin Class Link
    class Link {

        attribute rel[0-1] : String;
        attribute hrefl[0-1] : String;
        attribute type[0-1] : String;
        attribute hreflang[0-1] : String;
        attribute title[0-1] : String;
        attribute lenght[0-1] : Integer;
        reference atom : ATOM oppositeOf links;
        reference entry : Entry oppositeOf links;
    }-- End Class Link

-- Begin Class Person
    class Person {
        attribute name : String;
        attribute uri[0-1] : String;
        attribute email[0-1] : String;

    }-- End Class Person

-- Begin Class Author
    class Author extends Person {
    reference atom : ATOM oppositeOf authors;
    }-- End Class Author

-- Begin Class Contributor
```

```
      class Contributor extends Person {
      }-- End Class Contributor

--    class Date {
--          attribute day : Integer;
--          attribute month : Integer;
--          attribute year : Integer;
--          attribute hours : Integer;
--          attribute minutes : Integer;
--          attribute seconds : Integer;
--    }
}

package PrimitiveTypes {
      datatype String;
      datatype Integer;
}
```

II. Graphical Meta model of ATOM in UML



Semantic of transformations

RSS 2.0	Atom 1.0	Comments
rss	-	Vestigial in RSS
channel	feed	
title	title	
link	link	Atom defines an extensible family of rel values
description	subtitle	
language	-	Atom uses standard xml:lang attribute
copyright	rights	
webMaster	-	
managingEditor	author or contributor	
pubDate	published (in entry)	Atom has no feed-level equivalent
lastBuildDate (in channel)	updated	RSS has no item-level equivalent
category	category	
generator	generator	
docs	-	
cloud	-	
ttl	-	<ttl> is problematic, prefer HTTP 1.1 cache control
image	logo	Atom recommends 2:1 aspect ratio
-	icon	As in favicon.ico
rating	-	
textInput	-	
skipHours	-	
skipDays	-	
item	entry	
author	author	
-	contributor	
description	summary and/or content	depending on whether full content is provided

comments	-	
enclosure	-	rel="enclosure" on <link> in Atom
guid	id	
source	-	rel="via" on <link> in Atom
-	source	Container for feed-level metadata to support aggregation

See http://www.intertwingly.net/wiki/pie/Rss20AndAtom10Compared

Transformations

The project of transformation RSS to ATOM and ATOM to RSS contains 6 transformations:

1. XML to RSS
2. RSS to ATOM
3. ATOM to XML
4. XML to ATOM
5. ATOM to RSS
6. RSS to XML

All of this transformations was creates with ATL language.

XML to ATOM: permits to get an ATOM model from an “Ecore” file (model) XML. It requires the presence of two Meta models: ATOM and XML. The result will be stocked as an extension file “.ecore” (ATOM model).

ATOM to RSS: This transformation permits to get an RSS model from an “Ecore” file (model) ATOM. It requires the presence of two Meta models: ATOM and RSS. The result will be stocked as an extension file “.ecore” (RSS model).

RSS to XML: This transformation permits to get an XML model from an “Ecore” file (model) RSS. It requires the presence of two Meta models: XML and RSS. The result will be stocked as an extension file “.ecore” (XML model). The ***ecore*** file must be injected before obtain an XML file, this is the object.

XML to RSS: This transformation permits to get an RSS model from an “Ecore” file (model) XML. It requires the presence of two Meta models: XML and RSS. The result will be stocked as an extension file “.ecore” (RSS model).

RSS to ATOM: This transformation permits to get an ATOM model from an “Ecore” file (model) RSS. It requires the presence of two Meta models: ATOM and RSS. The result will be stocked as an extension file “.ecore” (ATOM model).

ATOM to XML: This transformation permits to get an XML model from an “Ecore” file (model) ATOM. It requires the presence of two meta models : XML and ATOM. The result will be stocked as an extension file “.ecore” (XML model). The ***ecore*** file must be injected before obtain an XML file, this is the object.

Example of transformation

I. ATOM model

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">

        <title>Example Feed</title>
        <subtitle>Insert witty or insightful remark here</subtitle>
        <link href="http://example.org/"/>
        <updated>2003-12-13T18:30:02Z</updated>
        <author>
                <name>John Doe</name>
                <email>johndoe@example.com</email>
        </author>
        <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

        <entry>
                <title>Atom-Powered Robots Run Amok</title>
                <link href="http://example.org/2003/12/13/atom03"/>
                <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
                <updated>2003-12-13T18:30:02Z</updated>
                <summary>Some text.</summary>
        </entry>

</feed>
```

II. RSS results model

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
        <channel>

                <title>Example Feed</title>
                <description>Insert witty or insightful remark
here</description>
                <link>http://example.org/</link>
                <lastBuildDate>Sat, 13 Dec 2003 18:30:02
GMT</lastBuildDate>
                <managingEditor>johndoe@example.com (John
Doe)</managingEditor>

                <item>
                        <title>Atom-Powered Robots Run Amok</title>
                        <link>http://example.org/2003/12/13/atom03</link>
                        <guid isPermaLink="false">urn:uuid:1225c695-cfb8-
4ebb-aaaa-80da344efa6a</guid>
                        <pubDate>Sat, 13 Dec 2003 18:30:02 GMT</pubDate>
                        <description>Some text.</description>
                </item>

        </channel>
</rss>
```