Community oriented system for information aggregation based on a template operated HTML parser

Ivan Kljaić, Jurica Ševa, Mark Čipčić
Faculty of Organization and Informatics
University of Zagreb
Pavlinska 2, 42000 Varaždin, Croatia
{ivan.kljaic, jurica.seva, mark.cipcic}@foi.hr

Abstract. *The definition of information crisis notices the problem of the information age: too much information which in return makes it hard to get to the needed information when needed. Almost every Internet user has experienced this effect of redundant amount of information delivered to him that can cause missing out on the needed information as well as repetition of already digested information. The gathering of information also includes data aggregation from more than one source which causes repetition of already seen information.*
In this paper we present a system that uses a user friendly method for the creation of website templates which define a HTML wrapper. Such system serves as information backbone for other developers which want to use data from other websites that do not offer any APIs. The possibilities of such a system are limitless for today's Semantic Web applications as well extended media Services

Keywords. HTML wrapper; XML tag; Semantic web application; Information exchange; Extended mediaIntroduction

1 Introduction

Today's web sites mostly represent Web applications which are composed of some kind of data, that are collected from a database and processed according to the user's needs, by a programming language (PL), and returned to the user. Every user has the possibility to create his own website (even for free) but professional businesses are web based and are organized around a large amount of data. The bigger ones have the need, the financial possibilities and resources to develop their own API-s (Application Programming Interface) and offer them to other users giving them a possibility to use the information available in the system. This increases the available information amount and is contributing to the grooving entropy of the information mass [8].

In this paper we present the concept of a system with the possibility for every user (personal users, developers, businesses) to access publicly available data and organize it based on his/hers needs by using a (user specific or node specific) Parsing [7] and filtering system that will access the available data. By using templates at the presentation level, the user would have access to information that are leaning more towards his/hers points of interest and would be able to make his own GUI. This system assumes the use of available open standards that are suggested by the W3C consortium and is following the guidelines of HTML 5 as the next standard for developing web applications. A high degree of authorisation control is evidently very important, because of the significance that every user receives the data that is filtered according to his needs. So the decision was easy to integrate a biometric authorisation system as mentioned in [3]

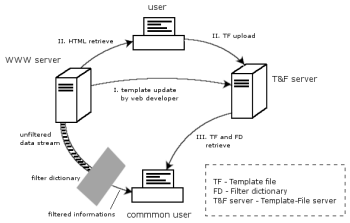

Figure 1: System model

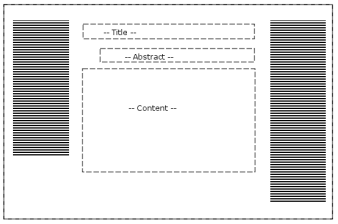

Figure 2: Template model

2 System model

The main structure of the system is as follows, as presented in 1:

- HTML node that acts as a carrier of the information
- HTML template that acts as the presentation level of the information
- DB system that holds the information about the user defined templates and filters
- programming language that analyses the information node
- user accessing the information source

The database holds following information:

- user defined templates
- user defined filters

The user creates his own template following a predefined HTML tag structure that serves as a presentation level of all the available information. This happens either by the website developer or by the common user which can share the template. If some organisation does not support his own data parsing to the system, the self organising nature of the common users could this way deliver the support as well as updates of the Template files.

Each user can add new, modify existing ones and delete filter rules that will allow for each of the available information nodes in the system to be evaluated based on his/her preferences and present information. The template here acts solely as a container for such information after parsing each (newly added) information node and displays them in the fashion defined by the user's template. The parsing rules are based on the available filtering system that is created using an underlying programming language that goes through the information node title, information node abstract and structure of the information node content. These filters are based on the combination of one or more phrases that the user defines both as favorable and unfavorable and are additionally supported by the filtering system logic in the background that will be explain the following chapter.

3 Basic concept

For this concept to work we assume that the HTML code of the parsed document will follow a structure specified by the template and that all users will follow the instructions of the model. The DOM (Document Object Modeling) guidelines set by the WWW Consortium allow the developers to define areas of each HTML document and to structure the content of each of the area that the document is comprised of. Such a way of structuring HTML documents (or a node in a broader sense of an element of the closed system) allow the parsing programs to go through the information part of the information node and gather needed descriptor from available information provided by the author of the content. The proposed structure of such a model is presented in 2.

The concept presented in this paper will be based on a template system (that are widely used by similar already available solutions like Joomla, Django [4], etc.) that will allow the user to insert the content in a predefined structured document with the key elements used by the parsing algorithms in the background to extract descriptors that will "position" the node in the information space. The main

elements of the information node DOM are suggested to be:

- information node title
- information node abstract that provides a brief summary of the information presented at the information node
- the full content of the information node

This template structure at the presentation level will make sure that all of the needed information to be inserted when adding new information to the system and will allow further parsing steps to be successfully implemented. The system uses user defined filter system, comprised of user entered phrases (one or more words), to reason if a specific node is, and if so to what extent, of interest to the user. The user can modify, add and delete filters with every visited node and add additional information that the parsing system in the background will use in further visited information nodes.

4 Filter system

The template system is the core of this concept and presents a way for the user to access specific site and filter out the newly added information that are or are not of his interest. Furthermore, by using additional tag system analysis, the user would have the possibility to receive filtered out information in a structured hierarchy based on the matching algorithm results which gives this system the real value.

When the user accesses an news source, the system will collect all the information created after the date of his last visit (available through the HTTP 1.1 [6] header meta option, e.g. <meta http-equiv="last-modified" content="Sun, 06 Nov 2005 14:59:42 GMT">;<meta http-equiv="if-modified-since" content="Sat, 05 Nov 2005 14:59:42 GMT">;) and analyse each found node that is created after that date [6]. The analysis will go through the title and abstract parts of the page and compare the content of those two sections with user defined filtering dictionary and based on the number of matches will position the information node higher or lower in the hierarchy of newly added information nodes. The filter system is presented in 3.

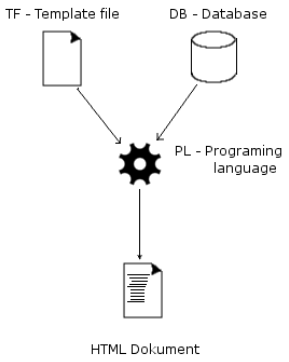

Figure 3: automatic HTML document creation

5 Community based template distribution

The idea behind the template (TF) distribution is to have a unified and centralised Website which would offer the user to distribute the temple. The template file for the website W can be uploaded by the website developer or the organisation which controls the website as well as any other common user that just wants to collect this data and review it other datasets from other websites the way he wants it.

The proposed system is handling template files inspired on Python's web development framework Django [4] which strongly separates application code from the HTML templates saved in a separate file. It is in essence an HTML document with breakpoints for the application. This way the templates could be handled by the HTTP (Hyper text transfer protocol) and saved in the local browser cache and use the ability of the "if-modified-since:" in the way that the community can upload a newer version of the template and the user that is requesting some informations has not to worry about template changes because of redesign of the website.

Such templates could also use other applications, as well as web applications what would have the consequence to give any simple developer the possibility to create new kinds of communication systems. The last thing according to the expected needs is the development of API-s and modules to different programming languages as our first goal is the full support for python [2].

6 The advantages of this system

There is a far more economic relationship to data storage resources as the information delivered to the user is only parsed and not recorded except the source server. It is a common problem that Websites often just copy the article and create redundant data on a global scope.

It is still not usual that the developers integrate RSS support. This way the user can ignore this problem and create the information supply for himself.

6.1 Solving the "information overflow" problem

As you seen from the previous examples how information can easily be extracted and displayed using a simple string filtering which the user had previous defined. We can only begin to imagine how we will more and more start to receive user specific and targeted informations.

For instance imagine the current situation if you are using an popular RSS [5] . After you have subscribed for the main news sources you favour, you will immediately start receiving news articles from these selected sources, there is no real "information overflow" problem occurring to a reader who daily reads new posts. But for instance if you have about 10 or more feeds that post in the frequency of 10 posts a day, and you don't get the time to watch and monitor all posts, the attention and relevancy for you of all the posts being collected is shrinking by each new entry, this is simply due to the simple fact that most people can not handle that amount of information in a limited amount of time.

To solve that problem, we decided that every user should simply be given the freedom to allow choosing and contributing what are the:

- most relevant keywords to the user
- most irrelevant keywords the user

In the extraction process of each node we have to analyse the given information source (node) and apply the given set of rules defined by each user, in a certain amount of data and time.

6.2 Websites that have no syndication (RSS)

Most visited and most popular content rich websites today surely have an RSS (real simple syndication) feed on their site. Each category contains a feed for itself. Our goal is to filter even in a given category the context and let the user decide what is important and what has a less degree of importance.

6.3 Information integration and displaying of information

On the user side the information displayed is highly valuable and targeted. The possibilities of such a system can then start to branch into new areas of interest. A user might get news feeds he expects to receive, but it's also very important that he can explore new areas the he isn't yet aware. We will evaluate user profiles based on similar keywords and similar points of interest.

In such a open source system users can and will contribute to make aggregation of information more useful and more enjoyable for them. The possibilities of such an open system are limitless. Take for example the case that user A can extract as previously explained information from n websites and having them displayed for him, now user A can automatically create and share his own feed which other users in the system.

6.4 Data integraton

With the semantic integration and HTML5 it's possible to achieve and retrieve more interrelating content from different sources like video, images, location, as well as text. Information can be more coherent as the evolution of such a changing system is continuing. We will continue to see more data integration that should be useful not only for human-human communication, but might also provide machine learning functionality. People today searching the Internet are trying to find good answers in sub-categories, the social element of this problem could be highly important as we go along in data integration

The importance of structured and standardised data in the future will become more important. Semantic web approach to data is one view only, but

the relationships among data is inevitably important, therefore data can be made more readable, understandable and linked.

7 Conclusion

Today's biggest problem is that humanity has started to create data with a far greater speed than to process and index them. Additional problems present them self in the manner that this data is not even connected and offers no structure so toady's approaches are leaning toward the unification of data relating to the semantic approach of data integration. Today's development frameworks poorly offer parsing techniques but also often do not offer development rules and guidelines that would prevent bad coding practise relating to HTML formatting. This system supports the creation of good coding practise guidelines as it offers any user to retrieve the information he really wants without the overwhelming amounts of data.

References

[1] Oram, A.; Wilson, Greg: Beautiful Code, O'Reilly, Sebastopol, USA, 2007.

[2] Jones A. C., Drake Jr. F. L.:Python & XML, O'Reilly, Sebastopol, USA, 2001

[3] BaČ C., Jurica Ševa, Ivan Kljaić : Ivan. E-authentication using chosen biometric characteristics; CECIIS, Conference Proceedings; Varaždin: Croatia, 2008:10: 24-26. pp. 401-406.

[4] Django Software foundation: The Django template language `http://www.docs.djangoproject.com/en/dev/topics/templates`, Accessed: 9th June 2010.

[5] RSS specification: `http://www.rss-specifications.com/rss-specifications.htm`, Accessed: 9th June 2010.

[6] W3C: HTTP 1.1 `http://www.w3.org/Protocols/rfc2616-sec14.html`, Accessed: 9th June 2010.

[7] Leonard Richardson: Beautiful Soup `http://www.crummy.com/software/BeautifulSoup`, Accessed: 9th June 2010.

[8] Patrick Ziegler: Three decades of data integration: All problems solved? `http://www.ifi.uzh.ch/stff/pziegler/papers/ZieglerWCC2004.pdf`, Accessed: 9th June 2010.