

INTERNET DATA ACQUISITION, SEARCH AND PROCESSING

by

Sandeep Neeli

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
Dec 18, 2009

Keywords: Data Mining, Citations, Data Processing

Copyright 2009 by Sandeep Neeli

Approved by:

Bogdan Wilamowski, Chair, Professor of Electrical and Computer Engineering
Thaddeus Roppel, Associate Professor of Electrical and Computer Engineering
John Hung, Professor of Electrical and Computer Engineering

Abstract

Internet data acquisition from the Web is the process of extracting essential data from any web server. Semi-structured data present in the form of HTML web pages need to be extracted, converted into structured data before presenting them to the users. In this thesis, four tools are presented which perform the functions - data acquisition, data search and data processing. They are: GradeWatch, Ethernet Robot, Online Search Tool and Citations Explorer Tool. GradeWatch is a tool mainly for students and faculty of an academic institution to check and post grades online respectively. Ethernet Robot extracts paper details for IEEE Transactions on Industrial Electronics from IEEE web server using Perl scripting language and processes the data using regular expressions.

Using the paper database created by Ethernet Robot, an Online Search Tool is developed which can perform a search up to a depth of three keywords and present the information on a separate web page from which the users can also download the papers by clicking the corresponding links provided with them. The Citations Explorer is a program which returns the most cited papers for the IEEE Transactions on Industrial Electronics for a particular year. The program uses Google Scholar to do the search and Perl regular expressions to process the data. The procedure for designing all these tools involves fetching, filtering, processing and presentation of required data. The resultant HTML files consisting of the required data are displayed for the perusal of users. Future enhancements to our Ethernet Robot include optimization to improve performance and customization for use as a sophisticated client-specific search agent.

Acknowledgments

I am heartily thankful to my supervisor, Dr. Wilamowski, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. I would like to thank my committee members, Dr. Hung and Dr. Roppel for accepting my request to be on my thesis committee. I would like to thank my family members for encouraging me to pursue this degree and Arthi Kothandaraman for providing me all the support. Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the thesis.

Table of Contents

Abstract	ii
Acknowledgments	iii
List of Illustrations	vi
1 Introduction	1
1.1 Phases of Automatic Web Data Extraction	1
1.2 Categories of Data used in Web Data Extraction	2
1.3 Current Trend	3
1.4 Pros and Cons	6
1.4.1 Pros	6
1.4.2 Cons	7
1.4.3 Engineering Constraints	7
2 GradeWatch	8
2.1 Overview	8
2.2 GradeWatch System Design	8
2.3 GradeWatch User Interface	10
2.4 Viewing Results	11
3 Data processing from IEEE Xlpore - Ethernet Robot	13
3.1 Overview	13
3.1.1 An Example	13
3.2 Data Collection	17
3.3 Data Filtering	17
3.4 Data Processing	18
3.5 Data Presentation	21

4	The online search tool	23
5	Citations Explorer Tool	26
5.1	Overview	26
5.2	Data Acquisition	27
5.3	Data Filtering	28
5.4	Data Processing	28
5.5	Data Presentation	29
6	Conclusions	30
6.1	Future Work	30
	Bibliography	31
	Appendices	34
A	Perl Source Code	35
A.1	Ethernet Robot - IEEE Web Data Extraction	35

List of Illustrations

1.1	Structure of various documents [1].	4
2.1	Data Flow in GradeWatch.	9
2.2	User Interface.	11
2.3	Course progress report of a student - user x.	12
3.1	IEEE Xplore webpage depicting various Data Fields.	14
3.2	Flowchart depicting the four stages of Ethernet Robot.	16
3.3	Output of wget implementation in Perl: Example.htm.	19
3.4	Execution of the Ethernet Robot Perl code.	20
3.5	Resultant web page.	21
3.6	Data Presentation.	22
4.1	Search Interface to download required papers.	23
4.2	Web page displaying the search results.	25
5.1	Sample page generated by Google Scholar.	27
5.2	Final web page for most cited papers for the year 2007.	29

Chapter 1

Introduction

The meteoritic rise of the World Wide Web as the knowledge powerhouse of the 21st century has led to a tremendous growth in information available to the masses. This, in turn, implies that the useful information is all the more time-consuming to narrow down or locate in the huge mass of available data. In other words, with increasing knowledge base, there is a pressing need to efficiently extract useful information in a shorter amount of time. Acquisition of structured data from a pile of unstructured documents is called Data Extraction (DE). Web data extraction is a process of extracting information or data from the World Wide Web (WWW) and manipulating it according to the user constraints. A brief overview of web data extraction is discussed below and we present an example model of web data extraction based on these features in the coming sections.

1.1 Phases of Automatic Web Data Extraction

The Web data extraction process can be divided into four distinct phases [20, 21, 26]:

1. Collecting Web data - Includes past activities as recorded in Web server logs and/or via cookies or session tracking modules. In some cases, Web content, structure, and application data can be added as additional sources of data.
2. Preprocessing Web data - Data is frequently pre-processed to put it into a format that is compatible with the analysis technique to be used in the next step. Preprocessing may include cleaning data of abnormalities, filtering out irrelevant information according to the goal of analysis, and completing the missing links (due to caching) in incomplete click through paths. Most importantly, unique sessions need to be identified from the different requests, based on a heuristic, such as requests originating from an identical IP address within a given

time period.

3. Analyzing Web data - Also known as Web Usage Mining [22, 23, 24], this step applies machine learning or Data Mining techniques to discover interesting usage patterns and statistical correlations between web pages and user groups. This step frequently results in automatic user profiling, and is typically applied offline, so that it does not add a burden on the web server.

4. Decision making/Final Recommendation Phase - The last phase in web data extraction makes use of the results of the previous analysis step to deliver recommendations to the user. The recommendation process typically involves generating dynamic Web content on the fly, such as adding hyperlinks to the last web page requested by the user. This can be accomplished using a variety of Web technology options such as CGI programming.

1.2 Categories of Data used in Web Data Extraction

The Web data mining process depends on one or more of the following data sources [25,26]:

1. Content Data - Text, images, etc, in HTML pages, as well as information in databases.
2. Structure Data - Hyperlinks connecting the pages to one another.
3. Usage Data - Records of the visits to each web page on a website, including time of visit, IP address, etc. This data is typically recorded in Web server logs, but it can also be collected using cookies or other session tracking tools.
4. User Profile - Information about the user including demographic attributes (age, population, etc), and preferences that are gathered either explicitly or implicitly.

The input file of a Data Extraction (DE) task may be structured, semi-structured, or free-text. As shown in Fig. 1.1, the definition of these terms varies across research domains. Free-texts, e.g., news article, that are written in natural languages are considered unstructured [28], postings on newsgroup (e.g., apartment rentals), medical records and equipment maintenance logs are semi-structured, while HTML pages are structured. According to the

database researchers [29], the information stored in computer databases is known as structured data; XML documents have the schema information mixed with the data values and hence, are semi-structured data. Web pages in HTML are unstructured because there is very limited indication of the type of data. XML documents are considered as structured since there is XML schema available to describe the data. Free texts are unstructured since they require substantial natural language processing. The huge quantity of HTML pages on the Web are semi-structured [30] since the embedded data are often exchanged through HTML tags. One source of these large semi-structured documents is from the deep Web, which includes dynamic Web pages that are generated from structured databases with some templates or layouts. For example, the set of book pages from ebay has the same layout for the authors, title, price, comments, etc. A page class is Web pages that are formed from the same database with the same template. Semi-structured HTML pages can also be generated by hand. For example, the publication lists from various researchers' homepages all have title and source for each single paper, though they are produced by different people.

1.3 Current Trend

Current tools that enable data extraction or data mining are both expensive to maintain and complex to design and use due to several potholes such as difference in data formats, varying attributes and typographical errors in input documents [1]. An Extractor or Wrapper is one of such tools, which can perform the Data Extraction and processing jobs. Wrappers are special program routines that automatically extract data from Internet websites and convert the information into a structured format. Wrappers have three main functions.

- Download HTML pages from a website.
- Search, recognize and extract specified data.
- Save this data in a suitably structured format to enable further manipulation [2].

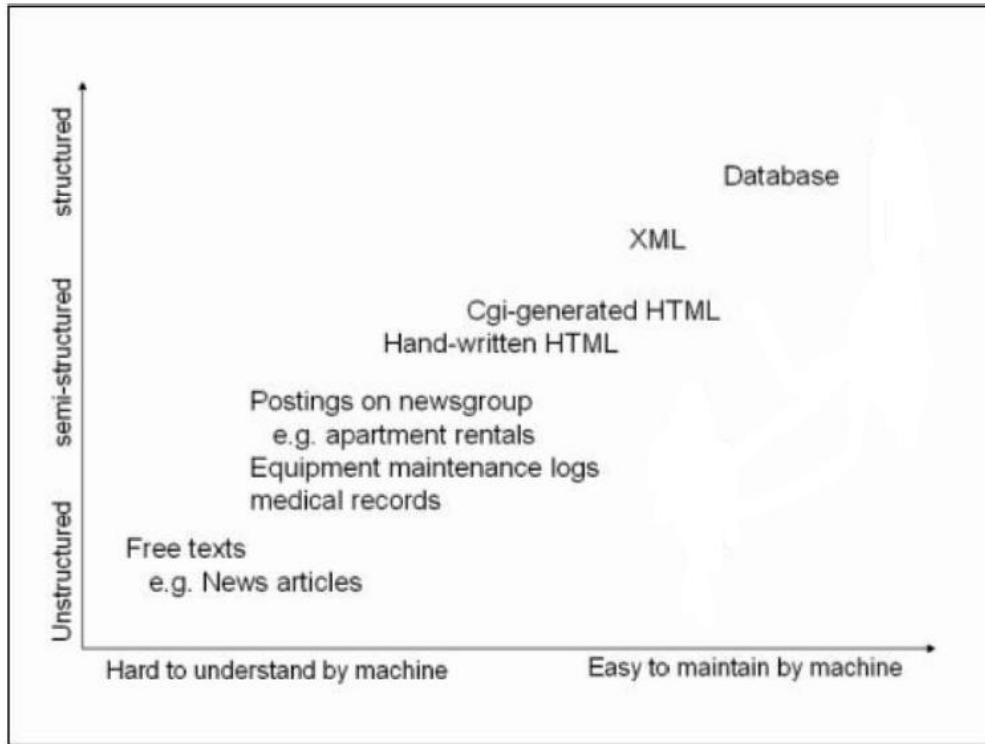


Figure 1.1: Structure of various documents [1].

The data can then be further imported into other applications for additional processing. Wrapper induction based on inductive machine learning is the leading technique available now a days. The user is asked to label or mark the target items in a set of training pages or a list of data records in one page. Using these training pages, the system learns extraction rules.

Inductive learning poses a major problem - the initial set of labeled training pages may not be fully depictive of the templates of all other pages. Poor performance of learnt rules is experienced for pages that follow templates uncovered by the labeled pages. This problem can be solved by labeling more pages, because more pages cover more templates. Despite, manual labeling requires a large supply of labor and is time consuming with an unsatisfied coverage of all possible templates. There are two main approaches to wrapper generation. The first and currently chief approach is wrapper induction. The second is automatic extraction. As discussed above, wrapper learning works as follows: The user first manually

labels a set of training pages or data records in a list. A learning system then generates rules from the training pages. Target items can be extracted from new pages by using these rules. Sample wrapper induction systems include WIEN [9], Stalker [10, 11, 12], BWI [13], WL2 [14].

An analytical survey on wrapper learning [15] gives a family of PAC-learnable wrapper classes and their induction algorithms and complexities. WIEN [9] and Softmealy [16] are earlier wrapper learning systems, which were later improved by Stalker [11, 10, 17, 12]. Stalker learns rules for each item and uses more detailed depiction of rules. It treats the items separately instead of ordering them. Though this method is more flexible it makes learning harder for complex pages as the local information is not fully utilized. Recent developments on Stalker are the addition of different active learning facilities to the system which has reduced the number of pages which a user needs to label. Active learning allows the system to select the most useful pages which a user labels and hence reduces manual effort [18].

Other tools typically used are roadrunner [3], WebOQL [4], Automated Data Extraction by Pattern Discovery [5], etc. Every day there is an exponential increase in the amount of information that seeps into the internet. Though this increases the possibility of finding a particular object, it also means a proportionate increase in search time. The tools for data extraction should therefore be developed with a view to reduce search time while keeping up with the internet advancements. In an attempt to serve this need, we present a new method of data extraction, called Ethernet Robot. Here, we make use of the Perl scripting language and the free non-interactive download utility- wget.exe.

Notable features of the Perl language, which form the core of our Ethernet Robot, is discussed below. Perl is the most prominent web programming language available because of its text processing features and developments in usability, features, and/or execution speed. Handling HTML forms is made simple by the CGI.pm module, a part of Perl's standard distribution. It has the capability to handle encrypted Web data, including e-commerce transactions and can be embedded into web servers to speed up processing by as much as

2000%. The function 'mod_perl' allows the Apache web server to embed a Perl interpreter [6]. Perl has a powerful regular expression engine built directly into its syntax. A regular expression or regex is a syntax that increases ease of operations that involve complex string comparisons, selections, replacements and hence, parsing. Regex are used by many text editors, utilities, and programming languages to search and manipulate text based on patterns. Regular expressions are widely used in our method to reduce the complexity of the code, to render the code obscure and powerful, and thus, unique. The combination of Perl, regular expressions and wget make Ethernet Robot an efficient solution for accelerated data downloading and extraction. Ethernet robot and its functionality are described in chapter 3. A complete description of GradeWatch, an online grade posting system is given in chapter 2. An online search engine based on the data extracted from Ethernet Robot is discussed in chapter 4. Finally, the online citations explorer tool, its operation and results are discussed in chapter 5 in addition to the conclusions and future work in chapter 6.

1.4 Pros and Cons

1.4.1 Pros

Web data extraction has many advantages for corporations and government agencies and hence they are the main users. This technology has enabled ecommerce to do personalized marketing, which eventually results in higher trade volumes. The government agencies are use this technology to analyze threats and fight against terrorism. The society can benefited by the predicting potential of this application by identifying criminal activities. The companies can establish better customer relationship by giving them exactly what they need. Companies can understand the needs of the customer better and they can react to customer needs faster. The companies can improve profitability by target pricing based on the profiles created.

1.4.2 Cons

Web Data Extraction technology when used on data of personal nature might cause concerns. The most criticized ethical issue involving web mining is the invasion of privacy. Privacy is considered lost when information concerning an individual is obtained, used, or disseminated, especially if this occurs without their knowledge or consent.

Another important concern is that the companies collecting the data for a specific purpose might use the data for a totally different purpose, and this essentially violates the users interests.

1.4.3 Engineering Constraints

Several websites on the Web do not allow robots to crawl through their web sites and grab information which reduce the performance of their system. One of the tools described in this thesis is Ethernet Robot which was used to extract data from the IEEE server systematically. IEEE has issued 'No Robots Policy' which states that downloading database or any portion of a publication's issue or volume in a systematic fashion is strictly prohibited. The use of robots or intelligent agents on their site is a violation of subscription license agreement. Creative solutions have to be developed to overcome these violations.

Chapter 2

GradeWatch

2.1 Overview

GradeWatch is a web-based database which allows students to check their advancement in courses they take using a web browser. Perl is used to design the interface and is integrated to the internet by a web server [36]. A student needs to be updated with his or her feedback gained through the given homework and projects. This helps in the learning technique of the student. Typically, different types of work and projects give away different weights to the final grade. So, it is difficult to keep the student notified about his or her current standing. One easier way is to use spreadsheets but it kills effort and time of both the instructor and the student. Therefore, the need for a database that would be easy to use by both an instructor and students is determined. Though there are numerous database systems for each university, they are only restricted to a limited group of authorized people or staff. A student generally cannot always access the database anytime online. Therefore, keeping in mind all the students who take the course, and keeping in mind a multi-browser compatible webpage, the GradeWatch database was created [31].

The database can be categorized accordingly as the instructor chooses and the system is easy to use both by the instructor and the student.

2.2 GradeWatch System Design

In the developing stages of software, the accessibility of the software to the client machine and server machine has to be resolved. When requested by the client machine, applets

are dispatched through the network and execution is performed entirely on the client machine. The applet would have to query a database server located at the same machine where the web site is built [35].

A secure HTTP can be used if data security is a priority. This avoids the need to encrypt data. The server executes instructions based on the given information and sends the results back to the local machine that made the request [32][33].

Fig. 2.1 shows the program component division and data flow in the application. The user interface is programmed in HTML enhanced with JavaScript. The data flow on the server is taken care of by a Perl script. The script accesses databases, verifies access authorization, and generates the report containing the grades of the student upto date and is sent back to the client machine as a web page [34].

A student can either use a webpage or call a CGI script to access the database. The CGI

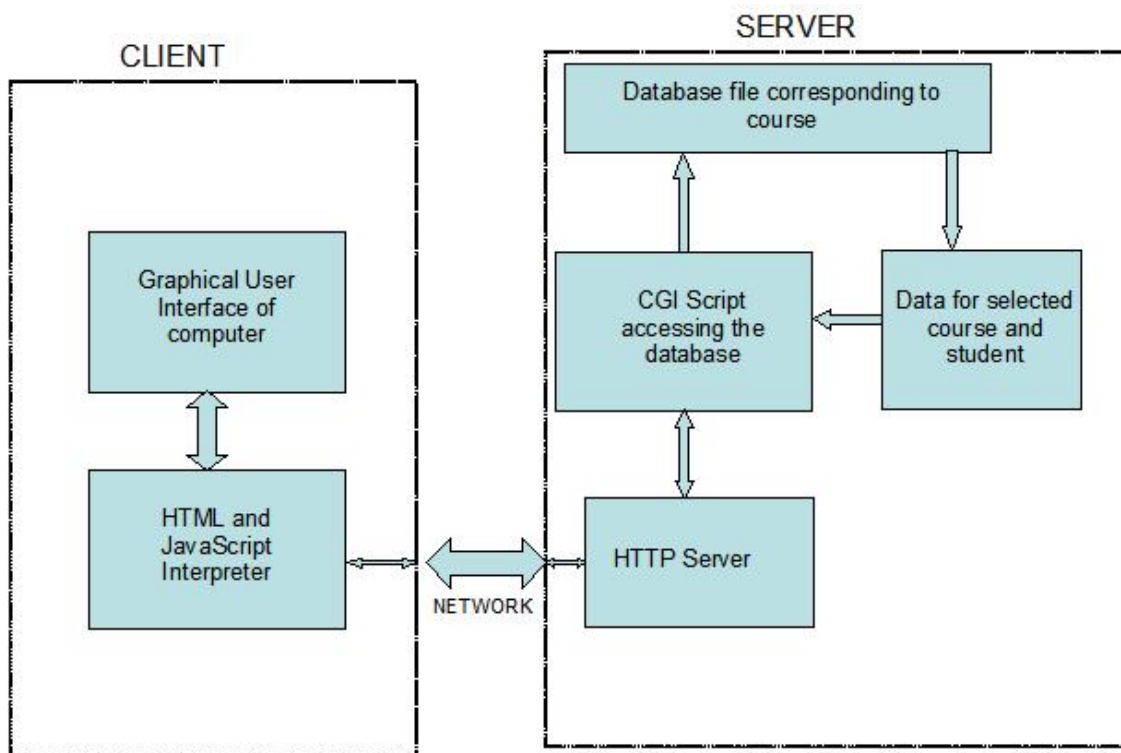


Figure 2.1: Data Flow in GradeWatch.

script generates a HTML page dynamically which consists of the list of courses with the

available databases. To retrieve the database, required details from a student are course number, his or her family name, and the student ID number as a password. The details are matched in accordance with the course selected. If the authorization fails, the user is denied access.

Other features of GradeWatch include:

1. Multiple accesses to the databases for the instructors.
2. E-mail notification to each student whenever there is an update in the spreadsheet/database.
3. Sending the grades individually to any specific student by e-mail.
4. Automatic setting up of a mailing list, which can be easily passed to any e-mail client program.

2.3 GradeWatch User Interface

To access the database, a person needs to provide the course number, family name, and student ID as a password. The database front-end is a web page with a CGI form as shown in Fig. 2.2. The CGI form takes the provided input from the user and the data is transferred to the server. The server which runs the CGI script receives the data as input. Perl is used to write the CGI script. First, the program verifies the data of a course in the list of courses and database of the instructor. Then the program processes the database for the corresponding course and retrieves the information required to produce a progress report. The report is produced as a HTML page or web page that is sent back to the web server, which redirects it to the client - a web browser. The authentic data shown relies on the arrangement of the grade database file for a specific course.

If an instructor desires to collect the database, more functionality is added. An instructor can access the grades of any student by entering the student's family name and the instructor's password, or can access all grades at the same time by providing the instructor's own name in the name field.

GradeWatch

COURSE:	<input type="text" value="MICROFAB"/> <input type="text" value="DEMO"/>
LAST NAME:	<input type="text"/>
PASSWORD:	<input type="password"/>
<input type="button" value="CONTINUE"/> <input type="button" value="clear"/> <input type="button" value="about"/>	

[Get your own grade posting system from here](#)
Send comments to wilam@ieee.org.

Figure 2.2: User Interface.

2.4 Viewing Results

In the sample database used for both, an illustration and as a demo on the Internet, the instructor's name and the password are both set to admin. After examining the report web page, the instructor has an option to send a grade report by e-mail to all students, to a particular student, or to a selected group. Optionally, a few lines of additional memo may be appended. A sample instructor's report for all students is shown in Fig. 2.3.

GRADES FOR MICROFAB

Name: USER X

Updated on Thursday, August 21, 08, 22:51

event	points	available	Other grades in the class			
			maximum	average	median	minimum
HW1	9.0	10.0	10.0	9.6	10.0	8.0
HW2	10.0	10.0	10.0	9.8	10.0	8.0
HWs	19.0	20.0	20.0	18.9	20.0	10.0
Exam1	96.0	100.0	100.0	95.1	97.0	78.0
grade	0.95	1.00	1.00	0.95	0.97	0.79

userx@auburn.edu

Figure 2.3: Course progress report of a student - user x.

Chapter 3

Data processing from IEEE Xlpore - Ethernet Robot

3.1 Overview

This section presents the implemented model of data extraction that can, in fact, draw only the necessary data from any web server on the internet. It can further be developed into a powerful search engine/portal. Typically, a Data Extraction (DE) task is well-marked by its input and its extraction target. The inputs are usually unstructured documents like the semi-structured documents that are present on the Web, such as tables or itemized lists or a free text that is written in natural language [7]. Our model of Data Extraction (DE), Ethernet Robot, can be used to download and extract any kind of information present on the internet according to the user requirements.

3.1.1 An Example

We consider an example of extracting titles and authors, pages, abstract URLs corresponding to the titles from the IEEE Transactions on Industrial Electronics located on IEEE Xplore. The main aim of this example is to allow the Associate Editors to search for reviewers, and Authors to search for paper references of the corresponding IEEE Transactions. The Transactions has papers listed according to the years of publication and each year has 6 issues. A screenshot of the transactions is shown in Fig. 3.1.

In this figure, the boxes indicate the required data to be extracted and the inessential data or junk to be filtered out from each issue. Lets now see how we automatically download and extract the data desired from these websites. Every Transactions on IEEE Xplore has a certain punumber, of which, Transactions on Industrial Electronics has a punumber = 41. The generalized URL of an issue Z for the year/volume no.- Y is given according to IEEE

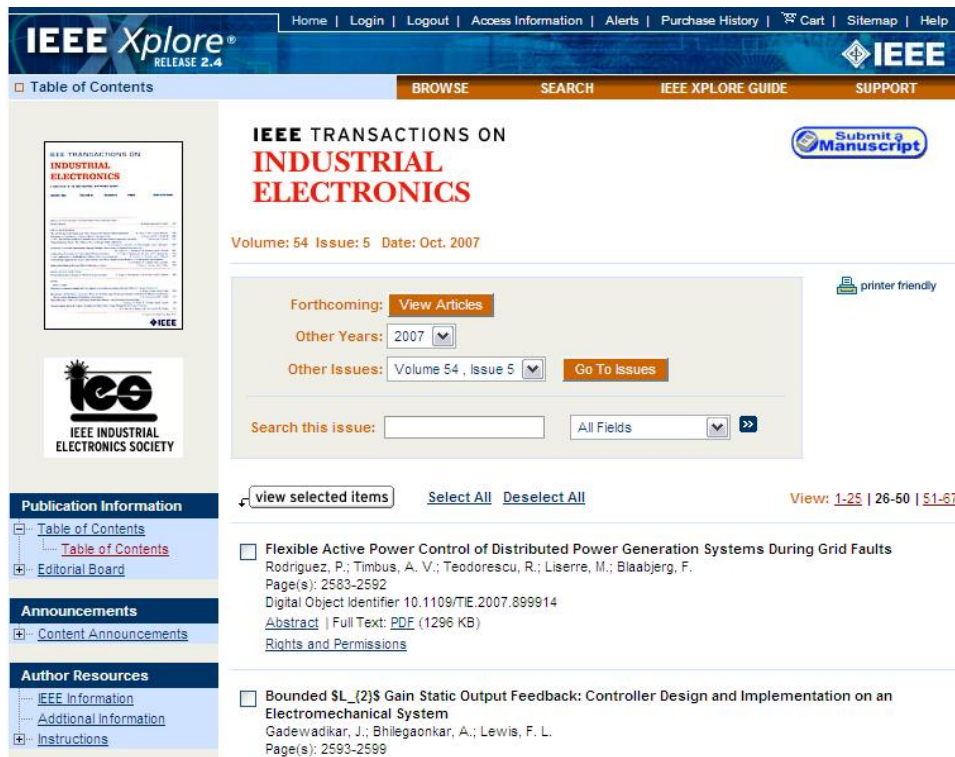


Figure 3.1: IEEE Xplore webpage depicting various Data Fields.

as:

<http://ieeexplore.ieee.org/servlet/opac?punumber=41&isvol=Y&isno=Z>

To download an extract the titles from volume no. 54 and issue 3 the URL is:

<http://ieeexplore.ieee.org/servlet/opac?punumber=41&isvol=54&isno=3>

Again each issue may have several pages 0,1,2,.. each page being addressed by :

<http://ieeexplore.ieee.org/servlet/opac?punumber=X&isvol=Y&isno=Z&page=P&ResultStart=Q>

where page = P denotes the page number P, ResultStart=Q denotes the start of title number Q.

The URL -

'<http://ieeexplore.ieee.org/servlet/opac?punumber=41&isvol=54&isno=3&page=1&ResultStart=25>'

is the link to the titles starting from number 26. The page P=0 of any issue contains the links/URLs of the remaining pages as shown in Fig. 3.1. So the other pages can be fetched using the wget function and can be concatenated to the page P=0 to form a single page containing all the paper listings. Following script does the above process:

```
$p = $p0.$p1.$p2;
```

where \$p0, \$p1 and \$p2 are the pages divided according to the paper listings and \$p denotes the webpage containing all the paper listings of an issue. So the behavior of the tool is defined by the variables volume no.-Y, issue number - Z, page number - P. To download all the pages from years 2000 to 2006 say, the following conditions have to be included at the beginning of main Perl code:

```
for($X=47, $x<=53, $x++)
{
for($Y=1,$Y<=6,$y++)
{
----
code
----
}
}
```

This example model of data extraction (Ethernet Robot) extracts all the titles and corresponding data from the IEEE Transactions on Industrial Electronics. In order to extract all data, the system needs to traverse all the paper list pages in the archive and then extract all the titles and data from each paper list page.

The code is devised to elicit the titles, authors, page numbers, abstract and abstract links from IEEE Xplore. Next, the Ethernet Robot goes to the webpage pointed by the URLs in each record and fetches the abstract. On completion of data acquisition, the raw data is printed in a new HTML file and published as a webpage.

The Ethernet Robot carries out four stages: Data collection, Data filtering, Data processing and Data presentation on web. A schematic representation of the sequence of steps is shown in Fig 3.2.

Of these, the data collection and filtering steps are relatively simple whereas the data pro-

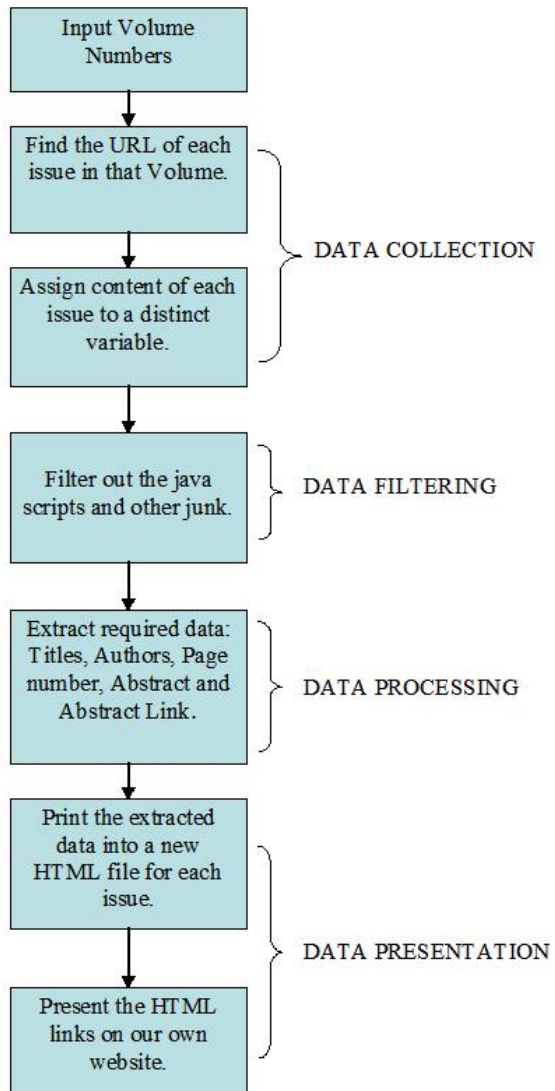


Figure 3.2: Flowchart depicting the four stages of Ethernet Robot.

cessing and presentation steps require more involved procedures. These steps are explained in greater detail in the following sections.

3.2 Data Collection

As mentioned in the previous section, the desired data to be fetched are specific volumes from the IEEE Transactions on Industrial Electronics. Thus, the starting point for this procedure is the Transactions webpage. Now, we invoke the function `get_page` with the parameter: volume number. `get_page` grabs the web pages corresponding to the volume number and returns one page per issue for each issue of the year/volume number.

The content of each issue is represented by a single variable `-$page`. Every year/volume has 6 issues, each of which is represented by a single element in an array of 6 variables. The following invariant holds true at any point during the operation of the code:

```
$p[$i]= $page for $i<=6
```

where `$p` - array of issue content

`$i` - issue number or iteration

`$page` - content of each issue

3.3 Data Filtering

Each webpage indicated by the variable `$p[$i]` consists of various irrelevant (to our purposes) JavaScript, html tags, tables and other miscellaneous information appended to the data we wish to extract. Hence, the content of the page needs to be filtered. The following condition in the code performs the proposed filtering operation:

where the new variable '`$entry`' holds the required content between the `<table>` and `</table>`.

```

while ($issuepage =~
m/<table[^>]*>\s*<tr[^>]*>\s*<td[^>]*>\s*(.*?)\s*</td>\s*</tr>\s*</table>/gis )
$entry = $1.

```

3.4 Data Processing

We now have the desired data in the variable \$issuepage. All we need is to extract them in an orderly fashion in accordance with the IEEE format of representation. Following condition using the regular expressions divides the variables \$1 through \$6 into titles, authors, abstract links and pdf file links:

```

if($entry =~
m/<strong>(.*?)</strong>\s*<br>\s*((.*?)\s*<br>)?\s*Page\s(\s\):&nbsp;\s.*<a\s
+href="(.*?)">.*<a\s+href="(.*?)">.*<a\s+href="(.*?)">/is)

```

which assigns the data into following variables:

\$title = \$1 - titles

\$authors = \$3 - authors

\$labs = \$4 - abstract links

\$lpdf = \$5 - pdf file links

We use wget.exe to obtain the abstracts from the abstract links. GNU Wget is a free utility for non-interactive download of files from the Web. It supports http, https, and ftp protocols, as well as retrieval through http proxies. Wget is non-interactive, which means that it can work in the background, while the user is not logged on. This allows the user to start retrieval and disconnect from the system, letting wget finish the work. In contrast, most of the Web browsers require constant user's presence, which can be a great hindrance when transferring a lot of data [8]. The operation of wget.exe can be explained briefly as below:

The implementation of wget in Perl is shown below to fetch the webpage addressing www.ieee.org:


```

use strict;

my $addr = "http://www.ieee.org/portal/site";
system("wget.exe", "-q", "-O", "example.htm", $addr);

```

The output of the above implementation is as displayed below in Fig. 3.3:

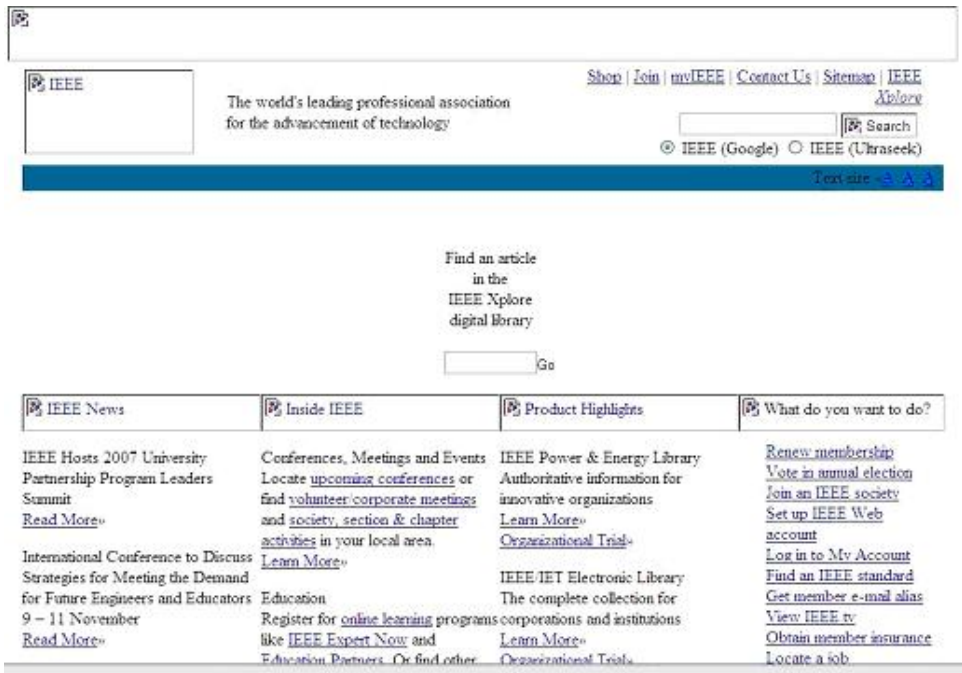


Figure 3.3: Output of wget implementation in Perl: Example.htm.

The essence of Ethernet Robot lies in the wget.exe function which downloads the desired data (Abstracts and pdf files) from the web pages. Wget.exe is responsible for the robotic behavior of our code as it does the automatic extraction of abstracts and pdf files.

The sub function get_page makes use of wget.exe to extract all the data from a webpage.

```

sub get_page
{
my ($addr) = @_;

```

```

$addr = s/& /& /g;
my $fname="54.1.htm";
system("wget.exe","-q","-O", $fname,"-referrer =http://tie.ieee-ies.org/tie/", $addr);
my $page=get_file($fname);
unlink($fname);
return($page);
}

```

The wget parameter "-O \$fname" specifies that the content in the webpage indicated by

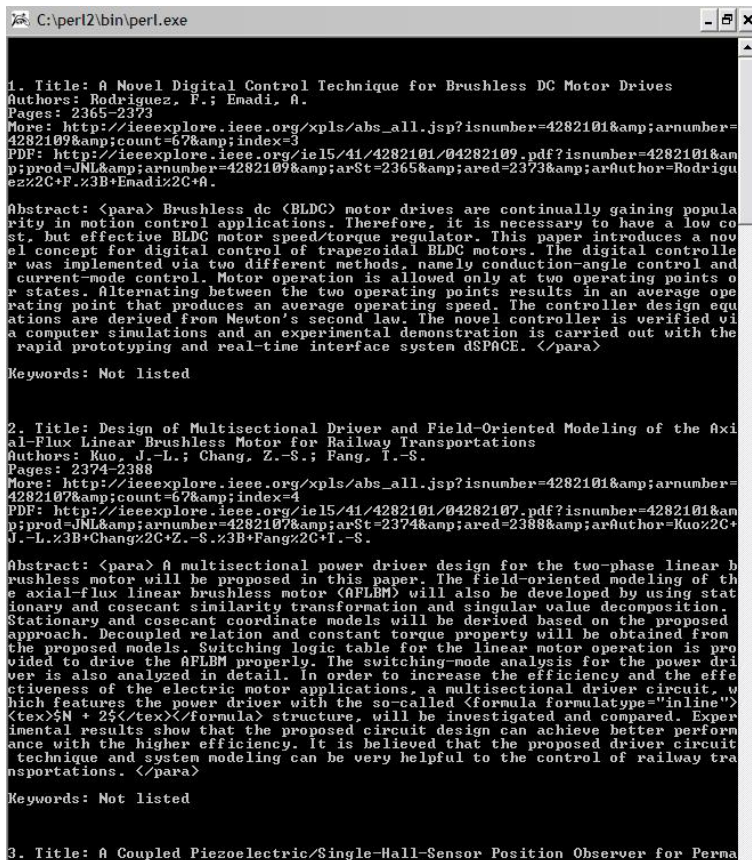
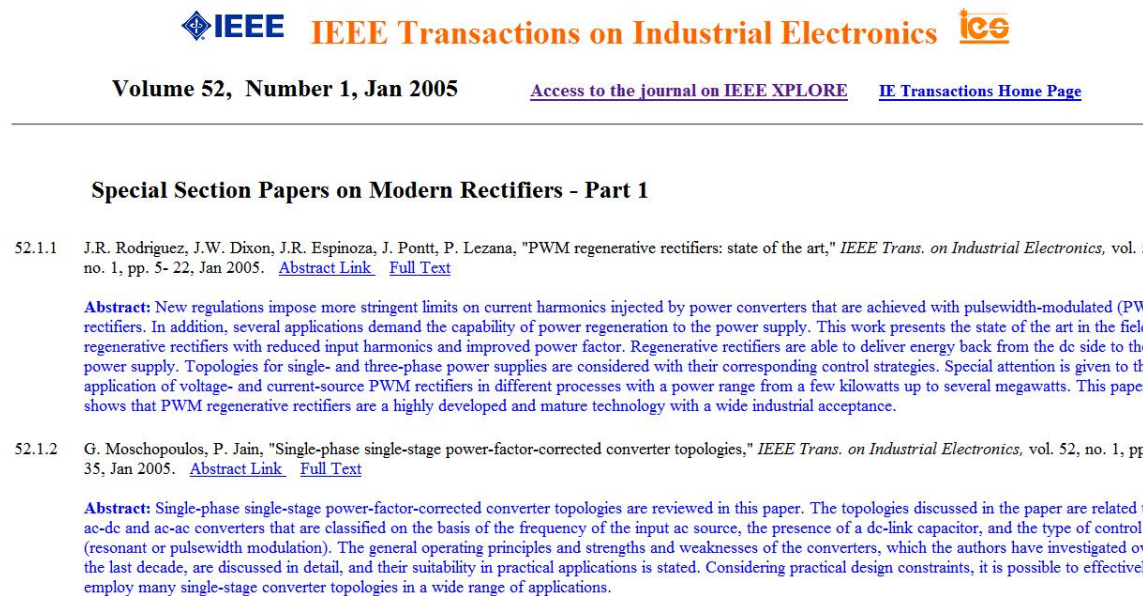


Figure 3.4: Execution of the Ethernet Robot Perl code.

\$addr will be printed to the file - \$fname i.e., 54_1.htm. Hence, at the end of execution of the code, we have all the required data corresponding to the issues in 6 separate files per year/volume. The execution of the Perl code is shown in Fig. 3.4.

3.5 Data Presentation

Processing of data obtained in the previous section results in a web page which contains all the papers with corresponding titles, authors, abstract links, pdf links with complete abstract. The resulted web page is as shown in Fig. 3.5.



The image shows a screenshot of the IEEE Transactions on Industrial Electronics website. At the top, there is the IEEE logo followed by the journal title "IEEE Transactions on Industrial Electronics" and the IES logo. Below this, it says "Volume 52, Number 1, Jan 2005" and provides links for "Access to the journal on IEEE XPLORE" and "IE Transactions Home Page". A horizontal line separates the header from the main content. The main content is titled "Special Section Papers on Modern Rectifiers - Part 1". There are two entries listed:

- 52.1.1 J.R. Rodriguez, J.W. Dixon, J.R. Espinoza, J. Pontt, P. Lezana, "PWM regenerative rectifiers: state of the art," *IEEE Trans. on Industrial Electronics*, vol. 52, no. 1, pp. 5- 22, Jan 2005. [Abstract Link](#) [Full Text](#)
Abstract: New regulations impose more stringent limits on current harmonics injected by power converters that are achieved with pulsewidth-modulated (PWM) rectifiers. In addition, several applications demand the capability of power regeneration to the power supply. This work presents the state of the art in the field of regenerative rectifiers with reduced input harmonics and improved power factor. Regenerative rectifiers are able to deliver energy back from the dc side to the power supply. Topologies for single- and three-phase power supplies are considered with their corresponding control strategies. Special attention is given to the application of voltage- and current-source PWM rectifiers in different processes with a power range from a few kilowatts up to several megawatts. This paper shows that PWM regenerative rectifiers are a highly developed and mature technology with a wide industrial acceptance.
- 52.1.2 G. Moschopoulos, P. Jain, "Single-phase single-stage power-factor-corrected converter topologies," *IEEE Trans. on Industrial Electronics*, vol. 52, no. 1, pp. 35, Jan 2005. [Abstract Link](#) [Full Text](#)
Abstract: Single-phase single-stage power-factor-corrected converter topologies are reviewed in this paper. The topologies discussed in the paper are related to ac-dc and ac-ac converters that are classified on the basis of the frequency of the input ac source, the presence of a dc-link capacitor, and the type of control (resonant or pulsewidth modulation). The general operating principles and strengths and weaknesses of the converters, which the authors have investigated over the last decade, are discussed in detail, and their suitability in practical applications is stated. Considering practical design constraints, it is possible to effectively employ many single-stage converter topologies in a wide range of applications.

Figure 3.5: Resultant web page.

The obtained files are then released into the World Wide Web by presenting them as links on a website. The files can be accessed by authors and editors, and reviewers for academic and professional use. A sample of the files shown on the web is shown in Fig. 3.6. The data present in the issues can be further processed using Perl to group all the issues of a year/volume in one whole file. Sample website which contains all the links for the desired data can be accessed from : <http://tie.ieee-ies.org/tie/abs/index.htm>



Papers in IEEE Trans. on Industrial Electronics

[IE Transactions home page](#)
 [IEEE Xplore](#)
 [IE Society home page](#)

Searching for authors and keywords

Open a link in your web browse (for example [abstracts 2006](#)) press Ctrl F key and type name or keywords. You will be able to search TIE papers for entire year of 2006

2006 to 2007 [titles](#) [abstracts](#)
2004 to 2005 [titles](#) [abstracts](#)
2000 to 2003 [titles](#) [abstracts](#)
1988 to 1999 [titles](#) [abstracts](#)

[The most recent issue](#)
[Forthcoming Articles - on XPLORE](#)
[Forthcoming Articles - Accepted papers](#)
[Forthcoming Articles - search friendly version](#)

2008	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6	issue7	issue8	issue9	issue10	issue11	issue12
2007	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
2006	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
2005	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
2004	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
2003	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
2002	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
2001	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
2000	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
1999	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
1998	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
1997	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
1996	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						
1995	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6						

Figure 3.6: Data Presentation.

Chapter 4

The online search tool

After successfully extracting the data and storing them in our database, a search interface has been developed which can actually display all the titles/papers for a set of three keywords taken from title or authors or any of the words in the abstract.

This tool is an addition to the Ethernet Robot, written in Perl/CGI which allows the user to search through the entire database we have extracted before and display the search results in a separate webpage. The search can be further refined or made selective by choosing the appropriate radio buttons for the corresponding years. Fig. 4.1 shows the search interface created for the above example.

DATABASE SEARCH (Returns All The Titles and Abstracts, Matching The Keyword)

Enter keyword I :

Enter keyword II :

Enter keyword III :

Please Select the Range: 2006-2007 2004-2005 2000-2004 1988-1999

Figure 4.1: Search Interface to download required papers.

This search tool can be included in the website hosted by our server by using Forms in the HTML page as follows:

```
<FORM METHOD="POST" ACTION="/cgi-  
bin/examples/search.cgi"  
.  
.  
</FORM>
```

In the above HTML script, ACTION tells the server to execute the search.cgi program on hitting the button Submit. The program search.cgi performs the data processing job again, fetching the files from the database on the server. Param, an inbuilt function in CGI script is used in this program to acquire user data from the HTML file.

For example, if the user mentions 'neural', 'networks' and 'motors' in the three query spaces-keyword I, keyword II and keyword III respectively, all the Titles and abstracts which have the keywords mentioned above will be displayed in a separate webpage. Fig. 4.2 shows the resultant webpage after performing the search operation. The user can right click the full text link to download and save the entire paper in pdf format. These links are provided by our server where the entire extracted database is present.

For the further development of this technique, we can create an online tool using the same core concept and Google search, which when searched for an author or a title gives the details such as - total number of papers, total number of citations, average number of citations per paper, average number of citations per author, average number of papers per author, average number of citations per year, the age-weighted citation rate. This tool can also be used to find out the most cited paper and the most downloaded paper.

SEARCH RESULTS

36. 2. 13. P.P. Fasang, "Analog/digital ASIC design for testability," *Trans. on Industrial Electronics*, vol. 36, no. 2, pp. 219-226, April 1989. [Full Text Link](#)

Abstract: The author addresses three issues in design for testability (DFT) for mixed analog/digital application-specific integrated circuit (ASIC) chips: controllability, observability, and completeness in testing are examined for commonly used analog functions, and the results culminate in an architecture for testable mixed analog and digital circuits. The architecture is designed to solve the problems associated with basic circuit configurations for different types of commonly used analog macros. Using the recommended architecture to gain access to control and observation test points in the analog portions of the mixed analog/digital ASIC, a series of analog test tables for several different analog functions have been derived. The analog test procedures are independent of any digital design for testability that might be used in digital portions of the ASIC. General testing procedures for current analog/digital ASICs are described along with desirable characteristics for testers for this type of circuit.

36. 2. 14. K.D. Wagner, T.W. Williams, "Design for testability of analog/digital networks," *Trans. on Industrial Electronics*, vol. 36, no. 2, pp. 227-230, April 1989. [Full Text Link](#)

Abstract: The testing of analog/digital integrated circuits is difficult since they allow direct access to relatively few signals. Since the probing of component pins is the fundamental chip production test technique possibly that of board test as well, i.e. in-circuit test, methods must be found to enhance the controllability and observability of internal signal networks. The authors provide a set of design for testability (DFT) principles that enhance their ability to test these networks when combined with the requisite analog test plans.

41. 2. 2. O. Vainio, S.J. Ovaska, "Tachometer signal smoothing with analog discrete-time polynomial estimators," *Trans. on Industrial Electronics*, vol. 41, no. 2, pp. 147-154, April 1994. [Full Text Link](#)

Abstract: The design of sampled-data polynomial estimators for noise reduction in industrial instrumentation applications is discussed. Unlike conventional lowpass filters, an estimator causes no delay on the polynomial-like primary signal. A general purpose design approach is described, incorporating notch frequencies for removal of narrow-band noise components, such as the 50/60 Hz line frequency. A 24-tap estimator is optimized for tachometer signal smoothing in motor control systems. An analog circuit architecture, targeted for silicon CMOS implementation, is described and simulated. <>

41. 4. 12. F.P. Dawson, R. Bonert, "High performance single-chip gating circuit for a phase-controlled bridge," *Trans. on Industrial Electronics*, vol. 41, no. 4, pp. 467-470, Aug. 1994. [Full Text Link](#)

Abstract: The increasing availability of single-chip low cost microcontrollers has made it possible to reconsider conventional hardware designs for a variety of gating circuits. This paper, in particular, presents design of a gating circuit for a six-pulse phase-controlled bridge utilizing a single-chip programmable microcontroller. The dynamic performance of the proposed gating circuit is similar to an analog circuit implementation. The resolution of the firing angle is better than 0.1 degrees at 60 Hz. Moreover, the system is designed to operate over a frequency range of 3 Hz to 120 Hz, and to automatically adapt to chaotic line frequency. The experimental verification of the performance criteria are also presented. Finally, an example of a special application for a dual-bridge AC to DC converter is presented.

Figure 4.2: Web page displaying the search results.

Chapter 5

Citations Explorer Tool

5.1 Overview

The Citations Explorer Tool is a program that retrieves and analyzes academic citations. The tool written in Perl uses Google Scholar to obtain raw citations, then analyzes these and presents the following statistics:

- Total number of papers
- Total number of citations
- Average number of citations per paper
- Average number of citations per year

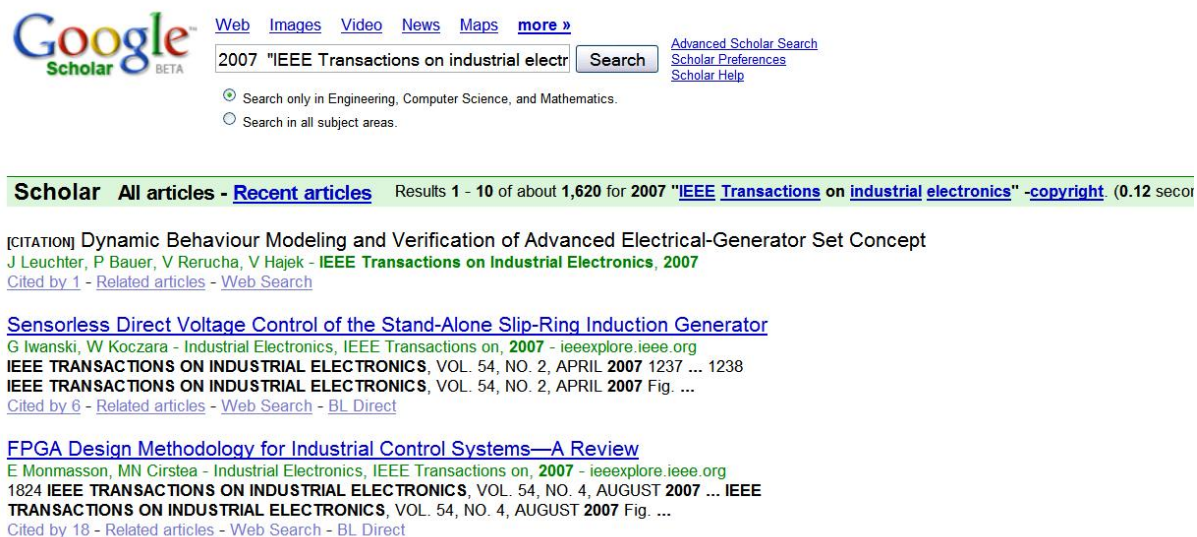
Design includes a code in Perl to search for most frequently cited IEEE TIE (Transactions on Industrial Electronics) papers in a specific year and displaying the obtained details of papers on a separate webpage.

The tool works similar to that of the Ethernet Robot we discussed in chapter 3 containing all the stages: Data Acquisition, Data Filtering, Data Processing and Data Presentation. Data acquisition is done by the Google Search API for Google Scholar. All the paper listings and their corresponding citation results are returned by the Search API in one web page. Data on this resultant web page are filtered and processed using regular expressions to present the data on a new web page.

5.2 Data Acquisition

Data acquisition is the core of this tool and is done by the Google AJAX Search API for Google Scholar. The Google AJAX Search API is a JavaScript library that allows us to embed Google Search in our web pages and other web applications. The Google AJAX Search API provides simple web objects that perform inline search over the Google service - Google Scholar. If a web page is designed to help users create content (e.g. citation analysis, message boards, blogs, etc.), the API is designed to support these activities by allowing them to copy search results directly into their messages.

The search query in our case is the year we want the citation results for followed by "Transactions on Industrial Electronics". The tool asks for the search query and for the year when executed.



The screenshot shows the Google Scholar search interface. At the top left is the Google Scholar logo. To its right are navigation links: Web, Images, Video, News, Maps, and more. Below these is a search bar containing the query "2007 IEEE Transactions on industrial electronics" and a Search button. To the right of the search bar are links for Advanced Scholar Search, Scholar Preferences, and Scholar Help. Below the search bar are two radio buttons: "Search only in Engineering, Computer Science, and Mathematics." (selected) and "Search in all subject areas." Below the search bar is a green bar with the text "Scholar All articles - Recent articles Results 1 - 10 of about 1,620 for 2007 IEEE Transactions on industrial electronics -copyright (0.12 seconds)". Below this are three search results, each with a title, author names, journal information, and citation links.

Scholar All articles - Recent articles Results 1 - 10 of about 1,620 for 2007 "IEEE Transactions on industrial electronics" -copyright (0.12 seconds)

[CITATION] Dynamic Behaviour Modeling and Verification of Advanced Electrical-Generator Set Concept
J Leuchter, P Bauer, V Rerucha, V Hajek - **IEEE Transactions on Industrial Electronics**, 2007
Cited by 1 - Related articles - Web Search

[Sensorless Direct Voltage Control of the Stand-Alone Slip-Ring Induction Generator](#)
G Iwanski, W Koczara - *Industrial Electronics, IEEE Transactions on*, 2007 - [ieeexplore.ieee.org](#)
IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 54, NO. 2, APRIL 2007 1237 ... 1238
IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 54, NO. 2, APRIL 2007 Fig. ...
Cited by 6 - Related articles - Web Search - BL Direct

[FPGA Design Methodology for Industrial Control Systems—A Review](#)
E Monmasson, MN Cirstea - *Industrial Electronics, IEEE Transactions on*, 2007 - [ieeexplore.ieee.org](#)
1824 IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 54, NO. 4, AUGUST 2007 ... IEEE
TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 54, NO. 4, AUGUST 2007 Fig. ...
Cited by 18 - Related articles - Web Search - BL Direct

Figure 5.1: Sample page generated by Google Scholar.

The resulting huge collection of information is returned in a web page, part of which is shown in Fig. 5.1. The data in the webpage is semi structured since the embedded data are rendered regularly via the use of HTML tags [30]. This semi structured data contains the required information such as number of citations, titles, author names, volume number, issue number, month and year of publication and the page numbers.

5.3 Data Filtering

The purpose of data filtering is to assist the user in finding that one necessary piece of information. One of the advantages of filtering is the increase in processing speed. Most data may seem rather random but one can usually find patterns. Based on these patterns we can build structures that will let a user answer a few simple questions. With this information we can then narrow down our data to return a small result set. The user can then quickly scan through this small set of data to find the one piece that they need. Filtering permits you to present varying views of the data stored in a dataset without actually affecting that data. Filter property represents a string that defines the filter criteria.

The above discussed webpage consists of various irrelevant (to our purposes) JavaScript, HTML tags, tables and other miscellaneous information appended to the data we wish to extract. Hence, the content of the page needs to be filtered. The HTML tags and tables can be removed or filtered using the code [Appendix 3]:

```
while ($resultpage =~  
m/<table[^>]*>\s*<tr[^>]*>\s*<td[^>]*>\s*(.*?)\s*</td>\s*</tr>\s*</table>  
/gis )  
  
$filtereddata = $1.
```

Here \$resultpage is the unfiltered page containing required data and \$filtereddata contains only the information required. The regular expression above selects only the information present between the HTML tables of the web page and places it in a variable \$filtereddata.

5.4 Data Processing

In Data Processing, the data is run through the algorithms and characteristics and variables are identified and categorized, thereby transforming the data into broader, more meaningful pieces of information. We now have the desired data in the variable \$filtereddata. All we need is to extract them in an orderly fashion in accordance with the IEEE format

of representation. Following condition using the regular expressions divides the variables \$1 through \$6 into titles, authors, abstract links and pdf file links [Appendix]:

```
if($filtereddata=~
m/<strong>(.*?)</strong>\s*<br>\s*((.*?)\s*<br>)?\s*Page\ (s\):&nbsp;.*<a\s+h
ref="(.*?)">.*<a\s+href="(.*?)">.*<a\s+href="(.*?)"/is)
```

which assigns the data into following variables: \$title= \$1 - titles \$authors= \$3 - authors \$labs = \$5 - abstract links \$lpdf = \$6 - pdf file links The data corresponding to number of citations is extracted using a regular expression matching "Cited by" for each paper.

5.5 Data Presentation

The final outcome of data processing is a web page containing the contents of the variables \$title, \$authors, \$labs, \$lpdf grouped together representing title, authors, abstract links and full text links respectively. Page numbers, volume number, issue number and number of citations for each paper are displayed accordingly. The webpage containing the most cited papers on Transactions on Industrial Electronics (TIE) for the year 2007 with some added HTML script is shown in Fig. 5.2.

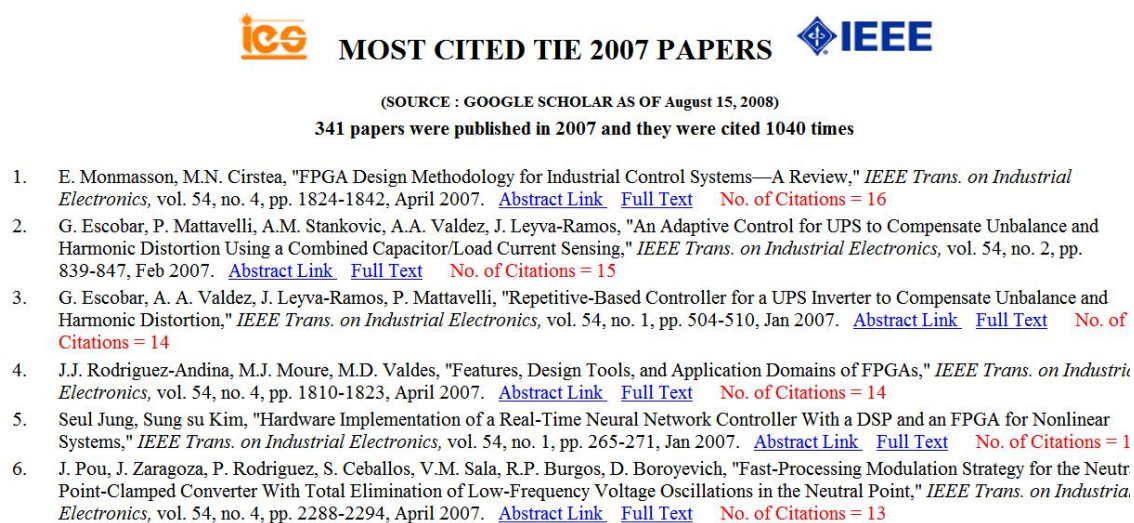


Figure 5.2: Final web page for most cited papers for the year 2007.

Chapter 6

Conclusions

The tool Ethernet Robot, delivers optimum performance, making it a unique tool for web data extraction. The tool stays within the constraints of 'IEEE No Robot Policy' and the information is extracted and presented on a different website only for the Associate Editors to search for reviewers, and Authors to search for paper references of the corresponding IEEE Transactions. GradeWatch is an efficient and platform independent grade posting system. The Online Search Tool provides accurate results for a given set of search keywords. The Citations Explorer Tool assisted by Google Scholar presents exhaustive list of papers and their citation count. The usage of Perl scripting language for all the tools, regular expressions and wget.exe make them accurate and advantageous. All the tools can be customized according to the required data and the format of the data which a user desires.

6.1 Future Work

Developments can be done in data extraction, processing and presentation levels. Extraction of data from the web pages can be improved by adding more pattern matching regular expressions. Intelligent data processing like making a title or statement automatically from a set of given keywords using genetic algorithms is possible and is a valid improvement to Ethernet Robot. Data flow between the server and client can be made simple by using database software like SQL or MySQL and scripting language PHP to access the SQL databases.

Bibliography

- [1] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis and Khaled F. Shaalan, "A Survey of Web Information Extraction Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1411-1428, October 2006
- [2] Wrapper Definition. http://www.knowlesys.com/articles/web-data-extraction/wrapper_definition.htm
- [3] Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," *Proc. the 26th Int'l Conf. Very Large Database Systems (VLDB)*, pp. 109-118, 2001.
- [4] G.O. Arocena and A.O. Mendelzon, "WebOQL: Restructuring Documents, Databases, and Webs," *Proc. 14th IEEE Int'l Conf. Data Eng. (ICDE)*, pp. 24-33, 1998.
- [5] C.-H. Chang, C.-N. Hsu, and S.-C. Lui, "Automatic Information Extraction from Semi-Structured Web Pages by Pattern Discovery," *Decision Support Systems J.*, vol. 35, no. 1, pp. 129-147, 2003.
- [6] About Perl. <http://www.perl.org/about.html>
- [7] I-Chen Wu, Jui-Yuan Su, and Loon-Been Chen, "A Web Data Extraction Description Language and Its Implementation", *Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05)*
- [8] GNU Wget 1.11.4 Manual. <http://www.gnu.org/software/wget/manual/wget.html>
- [9] Kushmerick, N.: Wrapper induction for information extraction. PhD thesis (1997) Chairperson-Daniel S. Weld.
- [10] Muslea, I., Minton, S., Knoblock, C.: A hierarchical approach to wrapper induction. In: *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*. (1999) 190-197
- [11] Muslea, I., Minton, S., Knoblock, C.: Active learning with strong and weak views: A case study on wrapper induction. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*. (2003)
- [12] Muslea, I., Minton, S., Knoblock, C.: Adaptive view validation: A first step towards automatic view detection. In: *Proceedings of ICML2002*. (2002) 443-450

- [13] Freitag, D., Kushmerick, N.: Boosted wrapper induction. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. (2000) 577-583
- [14] Cohen, W., Hurst, M., Jensen, L.: A flexible learning system for wrapping tables and lists in html documents. In: The Eleventh International World Wide Web Conference WWW-2002. (2002)
- [15] Kushmerick, N.: Wrapper induction: efficiency and expressiveness. *Artif. Intell.* (2000) 15-68
- [16] Hsu, C.N., Dung, M.T.: Generating finite-state transducers for semi-structured data extraction from the web. *Information Systems* 23 (1998) 521-538
- [17] Knoblock, C.A., Lerman, K., Minton, S., Muslea, I.: Accurately and reliably extracting data from the web: a machine learning approach. (2003) 275-287
- [18] Yanhong Zhai and Bing Liu. "Structured Data Extraction from the Web based on Partial Tree Alignment" Accepted for publication in *IEEE Transactions on Knowledge and Data Engineering*, 2006
- [19] IEEE Xplore OPAC Linking. <http://ieeexplore.ieee.org/xpl/opac.jsp>
- [20] Schafer J.B., Konstan J., and Reidel J. (1999). Recommender Systems in E-Commerce, In *Proc. ACM Conf. E-commerce*, 158-166.
- [21] Mobasher, B., Cooley, R., and Srivastava, J. (2000). Automatic personalization based on web usage mining, *Communications of the ACM*, 43(8) 142-151.
- [22] Spiliopoulou M. and Faulstich L. C. (1999). WUM: A Web utilization Miner, in *Proc. of EDBT workshop WebDB98*, Valencia, Spain.
- [23] Nasraoui O., Krishnapuram R., and Joshi A. (1999). Mining Web Access Logs Using a Relational Clustering Algorithm Based on a Robust Estimator, 8th International World Wide Web Conference, Toronto, 40-41.
- [24] Srivastava, J., Cooley, R., Deshpande, M., And Tan, P-N. (2000). Web usage mining: Discovery and applications of usage patterns from web data, *SIGKDD Explorations*, 1(2), 12-23.
- [25] Eirinaki M., Vazirgiannis M. (2003). Web mining for web personalization. *ACM Transactions On Internet Technology (TOIT)*, 3(1), 1-27.
- [26] Malinowski and B.M. Wilamowski, "Compiling Computer Programs Through Internet", *ITHET-2000 - International Conference on Information Technology Based Higher Education and Training*, Istanbul, Turkey, July 3-5, 2000, pp. 343-348.
- [27] Malinowski A. and B. M. Wilamowski, "Internet Accessible Compilers in Software and Computer Engineering" *International Conference on Simulation and Multimedia in Engineering Education (ICSEE'99)*, San Francisco, CA, January 14-15, 1999, pp. 221-224.

- [28] S. Soderland, "Learning to Extract Text-Based Information from the World Wide Web," Proc. Third Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 251-254, 1997.
- [29] R. Elmasri and S.B. Navathe, Fundamentals of Database Systems, fourth ed. Addison Wesley, 2003.
- [30] C. N. Hsu and M. Dung, "Generating Finite-State Transducers for Semi-Structured Data Extraction from the Web," J. Information Systems, vol. 23, no. 8, pp. 521-538, 1998.
- [31] Sweet, W. and Geppert, L., "http:// It has changed everything, especially our engineering thinking," IEEE Spectrum, January 1997, pp. 23-37.
- [32] Gundavaram, S., CGI Programming on the World Wide Web. O'Reilly & Associates, Inc., 1996
- [33] Jamsa, K.; Lalani, S.; Weakley, S.; Web Programming, Jamsa Press, Las Vegas, NV, 1996.
- [34] Wall,L., Christiansen, T., Schwartz, R.L. Programming Perl, O'Reilly & Associates, Inc., 1996.
- [35] Camposano, R.; Deering, S.; DeMicheli, G.; Markov, L.; Mastellone, M.; Newton, A.R.; Rabaey, J.; Rowson, J.; "What's ahead for Design on the Web", IEEE Spectrum, September 1998, pp. 53-63.
- [36] Wilamowski, Bogdan and Malinowski, Aleksander, "GradeWatch - the Software Package Displaying on Web Pages Students Grades", Proceedings ASEE Annual Conference 2000.

Appendices

Appendix A

Perl Source Code

The Perl programs to implement the tools presented in this thesis are presented here.

A.1 Ethernet Robot - IEEE Web Data Extraction

The source code for extracting paper details from IEEE Transactions on Industrial Electronics for the year 2007 is presented here.

```
-----Ethernet Robot(erobo.pl) - Perl Source Code-----
for(my $Y=53;$Y<=53;$Y++) #Access all issues of volume 53
{my $t =6;
  if($Y==(35||36)){ $t=4;}else{ $t=$t;}
for(my $X=1;$X<=$t;$X++) #Issue 1 to 6
{
    my $file = "C:/erobo/g/$Y-$X.htm";
open (HTM, ">", $file) or Error('open','file');
my $ad="http://ieeexplore.ieee.org/servlet/opac?
punumber=41&isvol=$Y&isno=$X";
my $issuepage= get_page($ad); #store the each issue webpage in a variable
open(KK,">", '22.htm');
my @rec = <KK>;
print KK $issuepage;
close KK;
open(KK,"<", '22.htm');
my @rc = <KK>;
close KK;
my @L=("","","","");my $k=0;
foreach my $l(@rc)
{
if(($l =~ m/ResultStart/i)&&(($l =~ m/page=1/i)||($l =~ m/page=2/i))) #concatenate all
the pages per issue into one variable
{
my $c="HREF=";
my $d="class";
$a=index($l,$d);
$b=index($l,$c);
$L[$k]=substr($l,$b+5,$c-33); $k++;

```



```

</h2><hr />";
my $count=0;
while ( $issuepage =~ m/<table[^>]*>\s*<tr[^>]*>\s*<td[^>]*>\s*
(.*?)\s*</td>\s*</tr>\s*</table>/gis ) # Filter out useful information
{
my $entry = $1;
if ($entry =~ m/<strong>(.*?)</strong>\s*<br>\s*((.*?)<br>)?
\s*Page\s*\(\s\):&nbsp;
\s*(\w+--\s*\w+).*<a\s+href="(.*?)".*<a\s+href="(.*?)".*<a\
s+href="(.*?)">/is) # Process out required values in to the
variables to be printed later
{
    $count++;
    my $title=$1;
    my $authors=$3; defined($authors) or $authors="";
    my $pages=$4;
    my $labs=$5;
    my $lpdf=$6;
    my $lcrt=$7;
    print $lpdf;
    print $lpdf;
    $title =~ s/\s+/ /g;
    $authors =~ s/\s+/ /g;
    ($labs =~ m/^http:\\\/\\\/) or $labs = "http://ieeexplore.ieee.org/" . $labs;
    ($lpdf =~ m/^http:\\\/\\\/) or $lpdf = "http://ieeexplore.ieee.org/" . $lpdf;
    ($lcrt =~ m/^http:\\\/\\\/) or $lcrt = "http://ieeexplore.ieee.org/" . $lcrt;
    my ($i, $name, $authors); # inverting orders in authors
    my @auth = split( /;/ , $authors);
    $authors = '';
    foreach (@auth)
    { (my $lasn, my $first) = split( /,/);
      $name = $first . ' ' . $lasn . ', ' ;
      $authors=$authors . $name;
      $authors =~ s/ / /g;
    }
    print OUT "\n$authors";
    my $abstract="";
    my $keywords="";
    my $pageabstract = get_page($labs );
    if (not defined($pageabstract))
    {
        $abstract="ERROR: Cannot access the abstract page";
    }
}
elseif ($pageabstract =~ m/<tr[^>]*>\s*<td[^>]*>\s*<span[^>]*>\
s*Abstract\s*</span>\s*<br>\s*(.*?)\s*</td>\s*</tr>/is)

```



```

close OUT;
open (DAT, $file);
my @rec = <DAT>;
close(DAT);
open(DAT,$file);
foreach my $line(@rec)
{
if($line=~ m/g><s/i)
{
my $word= "strong";
$line =~ s/\s*\b\Q$word\E\b(?!\.*\b\Q$word\E\b)//s;
my $word= "strong";
$line =~ s/\s*\b\Q$word\E\b(?!\.*\b\Q$word\E\b)//s;
my $word= "strong";
$line =~ s/\s*\b\Q$word\E\b(?!\.*\b\Q$word\E\b)//s;
#my $l=index($line,"</s"); print "FFFF= $l";
#substr($line,$l,$l)=' ';
}
#my $l=index($line,"<s");
#substr($line,$l,$l-50)=' ';
}
close(DAT);
open(DATA, ">", $file);
print DATA @rec;
close(DATA);
}
}
print HTM << "Bottom";
<p>&nbsp;</p>
</body>
</html>
Bottom
close HTM;
sub get_stdin
{
my $page="";
while (1) {
my $line=<STDIN>;
$line or last;
$page = $page . $line;
}
return($page);
}
sub get_file
{

```

```

my ($fname) = @_;
open (TMP, "<", $fname) or return(undef);
my $page="";
while (1) {
    my $line=<TMP>;
    $line or last;
    $page = $page . $line;
}
close(TMP);
return($page);
}
sub fun_month
{
my ($i) = $_[0];my $m = "";
if ($i==1){ $m = "Feb."; }
elsif ($i==2) {
$m = "April";
}
elsif ($i==3) {$m = "June";
}
elsif ($i==4) {
$m = "Aug.";
}
elsif ($i==5) {
$m = "Oct.";
}
else {
$m = "Dec.";
}
return($m);
}
sub get_page # Function to download full web page in to a variable
{
my ($addr) = @_;
$addr =~ s/&\/&/g;
my $fname="xxx.htm";
system("wget.exe", "-q", "-O", $fname, "--referer=
http://tie.ieee-ies.org/tie/", $addr);
my $page=get_file($fname);
unlink($fname);
return($page);
}

```