International Journal of Marketing and Technology

(ISSN: 2249-1058)

CONTENTS

Sr. No.	TITLE & NAME OF THE AUTHOR (S)	Page No.
1	**Customer Perception Towards Medical Tourism: The Next Big Leap**. Dr. V. Selvam, Dr. T.V. Malick and N. Abdul Nazar	1-18
2	**The Enigma Of Group Cohesion: An Empirical Study With Reference To Automobile Industry.** **Dr. V. Jayashree**	19-51
3	**Application of Soft Computing Techniques to Prediction of Faulty Classes in Object Oriented Software.** Divya Jain and Vibhor Sharma	52-66
4	**Ethical And Socially Responsible Marketing In Indian Business: A Conceptual Outline.** Lokendra Vikram Singh, Archana Singh and Dr. R.S. Waghela	67-89
5	**Rewards And Risks In Stock Markets: A Case Of Emerging Economies**. Pooja Yadav and Prof. B. S. Bodla	90-128
6	**Crawling, Indexing and Searching Silverlight Applications.** Ahmad Mateen and Muhammad Naveed Arshad	129-142
7	**Indian Economic Policy – Some Issues.** Dr. M. Sugunatha Reddy and Dr. B. Rama Bhupal Reddy	143-170
8	**A Study Focus On Working Capital Management Of Bharat Petroleum Corporation Limited (Bpcl).** Dr. N. PASUPATHI	171-187
9	**Consumer Attitude Towards Newspapers.** Mr. Ajit Dhar Dubey	188-207
10	**A comprehensive study about the Repercussion of Recession.** Richa Arora and Nitin R. Mahankale	208-224

Chief Patron

Patron

Chief Advisors

Editorial Board

Editorial Advisory Board

Associate Editors

Title

CRAWLING, INDEXING AND SEARCHING SILVERLIGHT APPLICATIONS

Author(s)

Ahmad Mateen

Department of Computer Science

University of Agriculture Faisalabad

Muhammad Naveed Arshad

Department of Computer Science

University of Agriculture Faisalabad

Abstract:

Current search engines are efficient for searching ordinary web pages. However, searching in today's evolving rich internet applications is difficult task for search engines. These evolving rich internet applications are adobe flash, adobe flex, Microsoft silverlight, Ajax and others are increasingly frequent on the web or mobile devices and are offering a high degree of interactivity to the user, by seamlessly loading content from the server without the need to refresh the page. Current search engines cannot perfectly search and index these applications. In this study we demonstrated that search engine for silverlight applications can be built. The main obstacle to achieve this goal was to extract textual information from silverlight objects, because silverlight objects are compiled into binary data packed into XAP file. These XAP files contain compiled DLLS. These DLLS contains islands of binary and textual data that can be read. This textual information is extracted and indexed is built.

Key Words: crawling silverlight websites, rich internet applications crawler, Search engines, silverlight crawler

Introduction:

Today, Search engines are primary source to search information worldwide web. Millions of pages are indexed on search engines. Large portion of World Wide Web can easily be searched easily. But on the other side lot of contents are still hidden from them. Evolution of new rich internet applications has revolutionized the web. Rich internet applications are much interactive, having rich user interfaces may contains motion graphics and having media playback features. Example of popular RIA includes Adobe Flash, Microsoft Silverlight and Ajax enabled applications.

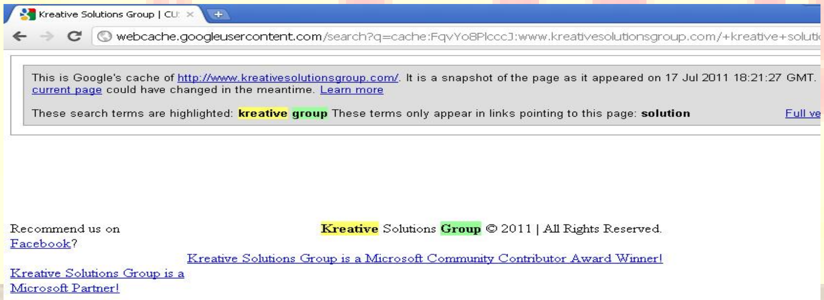
Current search engines cannot crawl them effectively. Some of them can be seen by search engine like Adobe flash. But some Rich internet applications like Microsoft silverlight could not be crawled yet.

Searching RIA is difficult job. They are not plain html. They are installed as browser plug-ins. These applications contains different types of objects like images, animations, transitions, forms, text fields and others makes it very difficult for search engines to perform their job effectively. A main issue related to RIA is to traverse them and get textual contents. Because this plug-ins is provided by different vendors, so, each has its own implementation structure.

Problem Domain:

Rich internet Applications including Silverlight are going to be popular. It has a high degree of interactivity and provides lot of functionality, which other applications are unable to provide. However Silverlight have great potential as a highly interactive, sophisticated candidate for next generation modern internet applications. According to Microsoft claimed 100 million Silverlight download in four weeks. In 2009 Microsoft claimed that 45% of internet connected machines have Silverlight installed.

Google, yahoo and other popular search engines fail to crawl and index web pages containing Silverlight objects. Silverlight is embedded in single webpage contains a lot of functionality under a single URL. These Silverlight objects are completely hidden from major search engines. Currently Search engines only index Meta tags information. We will use a website that contains silverlight object in our case study, named Kreative solution group.


Kreative Solutions Group | CU
webcache.googleusercontent.com/search?q=cache:FqvYo8PlcccJ:www.kreativesolutionsgroup.com/+kreative+soluti

This is Google's cache of http://www.kreativesolutionsgroup.com/. It is a snapshot of the page as it appeared on 17 Jul 2011 18:21:27 GMT.
current page could have changed in the meantime. Learn more

These search terms are highlighted: **kreative** **group** These terms only appear in links pointing to this page: **solution**

Full ve

Recommend us on Facebook?

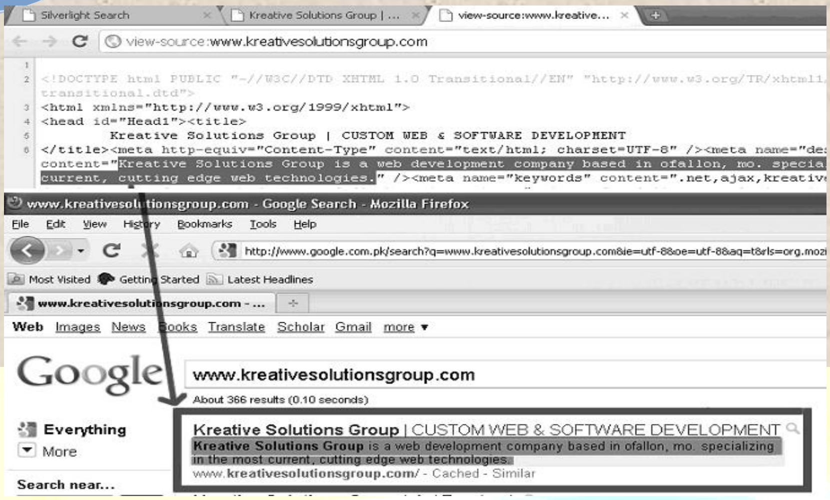
Kreative Solutions **Group** © 2011 | All Rights Reserved.

Kreative Solutions Group is a Microsoft Community Contributor Award Winner!

Kreative Solutions Group is a Microsoft Partner!

Figure 1 Cached version of Kreative solution group website in Google.

Currently google and other major search engines extract meta tags information as shown in Figure 1 and 2


Silverlight Search
Kreative Solutions Group | ...
view-source:www.kreative...
view-source:www.kreativesolutionsgroup.com
www.kreativesolutionsgroup.com - Google Search - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.google.com.pk/search?q=www.kreativesolutionsgroup.com&ie=utf-8&oe=utf-8&aq=t&rls=org.mozi
Most Visited Getting Started Latest Headlines
www.kreativesolutionsgroup.com - ...
Web Images News Books Translate Scholar Gmail more ▾
Google
Everything
More
Search near...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1"><title>
        Kreative Solutions Group | CUSTOM WEB & SOFTWARE DEVELOPMENT
</title><meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /><meta name="de
content="Kreative Solutions Group is a web development company based in ofallon, mo. specia
current, cutting edge web technologies." /><meta name="keywords" content=".net,ajax,kreativ
```

www.kreativesolutionsgroup.com

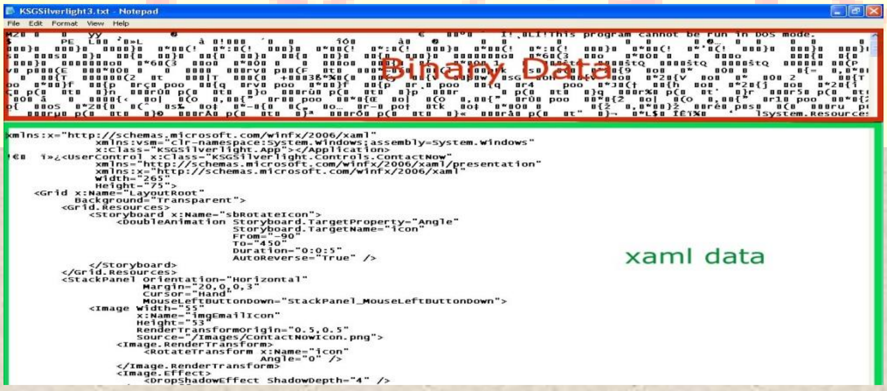
About 366 results (0.10 seconds)

Kreative Solutions Group | CUSTOM WEB & SOFTWARE DEVELOPMENT
Kreative Solutions Group is a web development company based in ofallon, mo. specializing in the most current, cutting edge web technologies.
www.kreativesolutionsgroup.com/ - Cached - Similar

Figure 2 Search engines only index meta information of silverlight applications.

Structure of silverlight applications:

Silverlight applications are deployed in form of XAP files. This XAP file is type of zip file that contains all files related to silverlight application to minimize and reduce network usage. All silverlight application is downloaded on client's machine at once. These XAP files contain XAML files. XAML files are actual files that contains silverlight application objects, like forms, graphics etc in form of XAML tags. These XAML tags are compiled into dll files. These dll files contain Binary data and XAML data. It is possible to extract text and links from these dll files. This is depicted in figure 3.



```
MZ      PE         !This program cannot be run in DOS mode.
                                                  system.Resources
```

```
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
          xmlns:vsm="clr-namespace:System.Windows;assembly=System.Windows"
          x:Class="KSGSilverlight.App"></Application>
<UserControl x:Class="KSGSilverlight.Controls.ContactNow"
          xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
          xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
          Width="265"
          Height="75">
    <Grid x:Name="LayoutRoot"
          Background="Transparent">
        <Grid.Resources>
            <Storyboard x:Name="sbRotateIcon">
                <DoubleAnimation Storyboard.TargetProperty="Angle"
                                 Storyboard.TargetName="icon"
                                 From="-90"
                                 To="450"
                                 Duration="0:0:5"
                                 AutoReverse="True" />
            </Storyboard>
        </Grid.Resources>
        <StackPanel Orientation="Horizontal"
                    Margin="20,0,0,3"
                    Cursor="Hand"
                    MouseLeftButtonDown="StackPanel_MouseLeftButtonDown">
            <Image Width="55"
                   x:Name="imgEmailIcon"
                   Height="53"
                   RenderTransformOrigin="0.5,0.5"
                   Source="/Images/ContactNowIcon.png">
                <Image.RenderTransform>
                    <RotateTransform x:Name="icon"
                                     Angle="0" />
                </Image.RenderTransform>
                <Image.Effect>
                    <DropShadowEffect ShadowDepth="4" />
```

Figure 3 DLL file containing binary data and XAML data

Architecture of Silverlight Search Engine:

An enhanced search engine capable of reading SILVERLIGHT applications has the architecture described below and is shown in Figure 4.

We solve the problem of crawling XAML files; we build a Silverlight spider, indexer and query processor, a crawler that can crawl through Silverlight objects. Like other Search engines it contains crawler, indexer and query processor. Our first attempt was to implement a minimum

Search engine that can crawl, index and can process user queries.

It works in the following way.

(i) Feed a crawl List.

(ii) Downloading silverlight website contains silverlight object.

(iii) Search cache to search and extract XAP files.

(iv)Unzipping and extracting dll files. These are contained within XAP files.

(v) Extracting textual contents, URLS, Text blocks and eliminating XAML tags.

(vi)Indexer performs different operation like stemming on contents and store contents in hash tables.

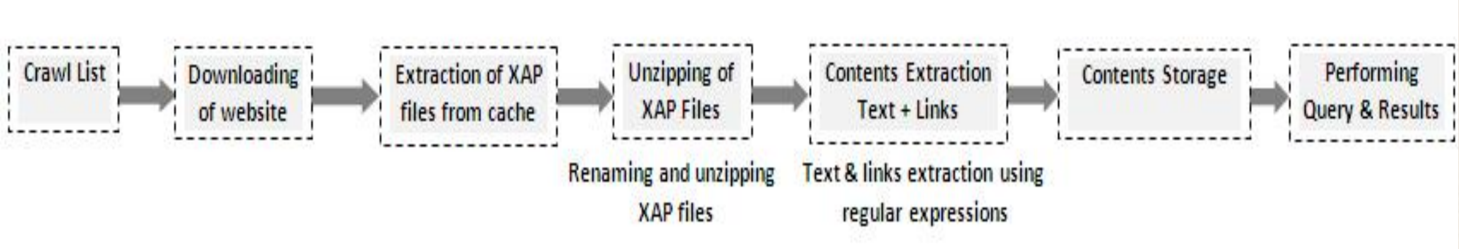
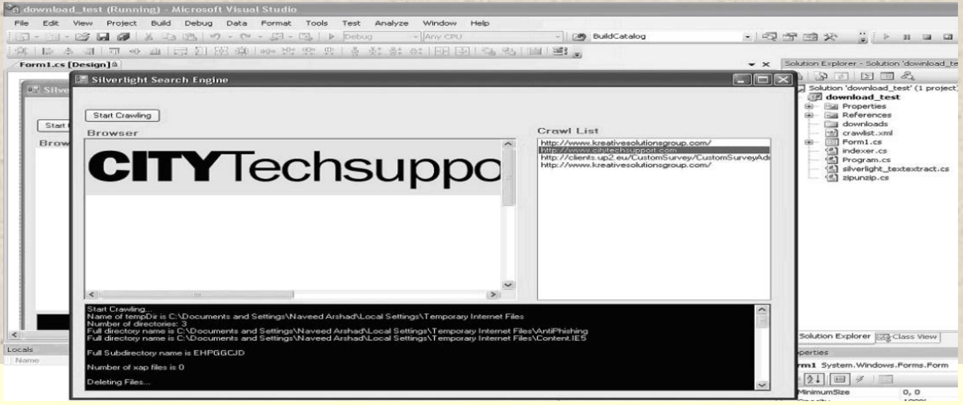

Crawl List
Downloading of website
Extraction of XAP files from cache
Unzipping of XAP Files
Contents Extraction Text + Links
Contents Storage
Performing Query & Results
Renaming and unzipping XAP files
Text & links extraction using regular expressions

Figure 4 Structure of Silverlight search engine

(vii) Perform search on search catalog.

Major components of silverlight search engine are described in detail below.


download_test (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Analyze Window Help
Debug
Any CPU
BuildCatalog
Form1.cs [Design]
Silverlight Search Engine
Start Crawling
Browser
CITYTechsuppo
Crawl List
Solution Explorer - Solution 'download_te
Solution 'download_test' (1 project)
download_test
Properties
References
downloads
crawlist.xml
Form1.cs
indexer.cs
Program.cs
silverlight_textextract.cs
zipunzip.cs
Solution Explorer
Class View
Locals
Name
System.Windows.Forms.Form
MinimumSize
0, 0

```
http://www.kreativesolutionsgroup.com/
http://www.citytechsupport.com
http://clients.up2.eu/CustomSurvey/CustomSurveyAd
http://www.kreativesolutionsgroup.com/
```

```
Start Crawling...
Name of tempDir is C:\Documents and Settings\Naveed Arshad\Local Settings\Temporary Internet Files
Number of directories: 3
Full directory name is C:\Documents and Settings\Naveed Arshad\Local Settings\Temporary Internet Files\AntiPhishing
Full directory name is C:\Documents and Settings\Naveed Arshad\Local Settings\Temporary Internet Files\Content.IE5

Full Subdirectory name is EHPGGCJD

Number of xap files is 0

Deleting Files...
```

Figure 5 Silverlight crawler in action, extracting XAP files.

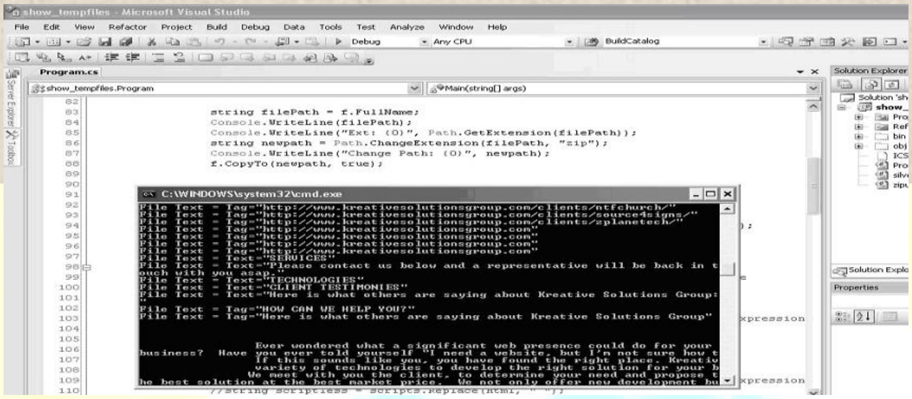
The Silverlight Crawler:

The function of silverlight crawler is to extract textual contents, URLs and other useful information from embedded object. Extracting textual data involves downloading xap files, unzipping it, extracting dll files then eliminating xaml tags and other binary data, extracting texts and links as shown in Figure 5.

To explain the crawling model we need to understand the Silverlight application and storage of Silverlight in browser's cache. A typical Silverlight application is mainly composed of XAP files, CSS (Cascading Style Sheets) files, html files and JavaScript files. But main Silverlight application is composed of XAP files. Mostly there are two XAP files, one is Silverlight preloader XAP file and other is Silverlight contents XAP file. First preloader runs and then loads actual file. XAP file is a zip file contains application manifest, configuration and dll files. These dll files are actually islands of XAML tags and binary data. These XAML tags can be accessed.

Indexer:

The indexer build index, it stores two types of files, XAML files. One file is for storing word file and other file is for storing files data. Each file has the following attributes, File URL, Title, Meta tags, crawled date and time and textual data. It is shown in Figure 6. The indexer has additional functionality like others standard search engine indexers. It performs stemming of words, exclude stop words and include go words. After Indexing each word and each file is stored and formatted in xml format. This xml file is formatted and serialized in binary data to reduce size then it is stored on hard-drive. This stored file can be used later for crawling purpose..


show_tempfiles - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
BuildCatalog
Program.cs
show_tempfiles.Program
Main(string[] args)
Solution Explorer
Properties
C:\WINDOWS\system32\cmd.exe

```
            string filePath = f.FullName;
            Console.WriteLine(filePath);
            Console.WriteLine("Ext: {0}", Path.GetExtension(filePath));
            string newpath = Path.ChangeExtension(filePath, "zip");
            Console.WriteLine("Change Path: {0}", newpath);
            f.CopyTo(newpath, true);
```

```
File Text = Tag="http://www.kreativesolutionsgroup.com/clients/ntfchurch/"
File Text = Tag="http://www.kreativesolutionsgroup.com/clients/source4signs/"
File Text = Tag="http://www.kreativesolutionsgroup.com/clients/zplanetech/"
File Text = Tag="http://www.kreativesolutionsgroup.com"
File Text = Tag="http://www.kreativesolutionsgroup.com"
File Text = Tag="http://www.kreativesolutionsgroup.com"
File Text = Tag="http://www.kreativesolutionsgroup.com"
File Text = Text="SERVICES"
File Text = Text="Please contact us below and a representative will be back in t
ouch with you asap."
File Text = Text="TECHNOLOGIES"
File Text = Text="CLIENT TESTIMONIES"
File Text = Text="Here is what others are saying about Kreative Solutions Group:
"
File Text = Tag="HOW CAN WE HELP YOU?"
File Text = Tag="Here is what others are saying about Kreative Solutions Group"


                Ever wondered what a significant web presence could do for your
business?  Have you ever told yourself "I need a website, but I'm not sure how t
                If this sounds like you, you have found the right place. Kreativ
                variety of technologies to develop the right solution for your b
               We meet with you the client, to determine your need and propose t
he best solution at the best market price.  We not only offer new development bu
```

Figure 6 Silverlight indexer showing text only.

Query Processor:

Query processor is used to process queries. The stored word index file is used to process queries. If index catalog is found on disk, it will be de-serialized and then loaded to memory. The query processor trims all white spaces, then split user query term by term. Each term is checked for go word and stemming. If it is go word then it is converted into lower case. If it is on go word then we removes

Unwanted characters and stemming is applied. The terms are loaded into hashtables

It is web based interface implemented in asp.net. User enters search query and submit to server, some operations are performed before query processing. Using regular expressions multiple white spaces are trimmed to one white space, and then if it is a Go Word then it will lower it. If it is not a Go word then apply stemming. Before stemming, punctuation characters are replaced. Before starting actual working current date and time is noted. Then results are stored in hash tables.

During searching results having higher keyword density will be shown on the top, Results are sorted according to keywords density order in descending order. Results title is title of web page containing search query. Description of a single result set contains have the following format, {Page Tile} {website name} {:}{ Title of Page Links…}. Results are divided into number of pages i.e ten results per page.

Crawling Challenges:

There are many crawling challenges we face during crawling. The first challenge is how to extract textual information from Silverlight application. We did extensive study to reach this objective. A major challenge is to properly download XAP files, We used a web browser control in application that reads web addresses from xml file to be crawled iteratively, So there is no proper mechanism to detect page loading completion, A problem with Silverlight applications that they contains preloader created by Microsoft, After preloading main Silverlight application loaded, So, when browser loading complete event occurs then we start traversing, extracting XAP files from cache, But files are not loaded completely, So when we extract XAP (XAP is actually a rename to zip file) file, It fails to extract. And our crawlers failed to proceed.

Another big challenge with Silverlight files that, the whole website is a single application, This creates a lot of problem, Crawler will unable to separate contents for different pages, Because all pages are merged into a single large object, So It is very difficult to distinguish contents for each page. If we succeed to distinct contents page wise then It is impossible to generate link of that page because Silverlight application loaded under single URL. Some Silverlight websites contains URLs with query string.

Conclusion:

Silverlight search engine is designed as a first attempt to show that Silverlight contents can be crawled and indexed. The primary goal of this search engine is to show contents from rich internet applications that are currently hidden from commercial search engines. We have shown a practical implementation of this concept.

Future Work:

Silverlight search engine is complex and have lot of issues and improvements needed for high quality search results. One quick workaround is index and Silverlight applications page-wise.

This has some limitations imposed by structure of Silverlight application developers. If they use query strings to link pages then pages can be indexed easily. Like commercial search engines, many other improvements can be applied like better page ranking, analysis of link structures, click through rates, inbound links analysis and many others. Storage of index in a well structured way. Our current structure is not efficient for large storages. And parallel crawler needs to be implemented for fast crawling and to improve throughput.

References:

- Duda C., G. Frey, D. Kossmann and C. Zhou (2008); “AJAXSearch: Crawling, Indexing and Searching Web 2.0 Applications”, *PVLDB '08, August 23-28, 2008, Auckland, New Zealand*
- Bar-Yossef Z. and L. Mashiach (2008); Local Approximation of PageRank Reverse PageRank, *SIGIR’08, July 20–24, 2008, Singapore. ACM 978-1-60558-164-4/08/07.*
- Brin S. and L. Page (1999); The Anatomy of a Large-Scale Hypertextual Web Search Engine, *Computer Science Department, Stanford University, Stanford, CA 94305*
- Grimes C. and S. O'Brein. (2008); Microscale Evolution of Web Pages, *WWW 2008, April 21–25, 2008, Beijing, China. ACM 978-1-60558-085-2/08/04.*
- Henzinger M. and S. Lawrence (2004); Extracting knowledge from the World Wide Web, *Google, Inc., 2400 Bayshore Parkway, Mountain View, CA 94043*
- Madhavan J., D. Ko, L. Kot, V. Ganapathy, A. Rasmussen and A. Halevy (2008); Google’s DeepWeb Crawl, *ACM. VLDB ‘08, August 2430,2008, Auckland, New Zealand*
- Madhavan J., D. Ko, L. Kot, V. Ganapathy, A. Rasmussen and A. Halevy (2008); Google’s DeepWeb Crawl, *ACM. VLDB ‘08, August 2430,2008, Auckland, New Zealand*