Accessing, Analyzing and Linking Data from DBLP with Other Internet Resources

Erhan Uyar, Sven Brehmer, and Malek Athamnah

***Abstract*— in this paper we present an approach to accessing, analyzing and linking data contained in the DBLP database. First, a general introduction to DBLP is given. Then, we briefly summarize related work. The subsequent part of the paper is devoted to explaining how interfaces of the DBLP database can be used and results from queries can be connected in order to make the returned data more expressive. Finally, we summarize our work and suggest topics for further research.**

I. Introduction

EVALUATION of data from the web not only brings numerous advantages, but also disadvantages. One of the main disadvantages is having lots of data referring to the same object without any connection between them. Therefore need for a linked data is required and needed in order to find the needle in the hay stack.

For a researcher it is very important to find related publications, authors, journals or conferences closer to their research area. The problem here is the difficulty of finding this information, since there is a big amount of content on the web. Researchers can run many queries on most popular databases on their research field; however those results will not be connected with each other.

Lucky for us, researchers in Computer Science field, there is a database called DBLP, which provides some of these connections and helps researchers to reach their goal faster. However, we could not find an application that puts the data in this data store into a satisfying context. Frontends for DBLP merely list requested data and leave further data inquiry to the user. Thus, our motivation is to create an application that not only retrieves records from the DBLP database, but also finds and displays related objects that may be of interest to the user. Furthermore, we explore possibilities for presenting related data in a useful manner (e.g. statistical analysis).

A. What is DBLP?

DBLP, short for "Databases and logic programming", is a database that contains metadata on scientific papers and related information. It is maintained by the University of Trier in Germany. According to the creator of DBLP, Michael Ley, the system does not rely on DBMS software, but rather utilizes "[...] a small set of programs and scripts written in C, Perl, Shell, and Java [...]" and "[...] the MG information retrieval system described in the excellent 'Managing Gigabytes' book [...]" [1]. The data records are stored in one big XML file.

Although its main purpose is to collect and provide bibliographic information on publications, it is also used for other reasons. On one hand, this includes the evaluation of new algorithms that analyze and parse XML files, while not paying attention to the semantics of the actual data. On the other hand, it is used for the generation of graphs and other statistical evaluations based on its content [2].

1) DBLP interfaces

DBLP allows users to perform searches based on different attributes. However, the official website [3] only allows search for author names. Third party websites like Faceted DBLP [4] provide additional capability, such as search for venue, year, publication type or scanning terms in all of the metadata.

An alternative use to the above-mentioned GUI interfaces for searches is downloading the DBLP database for local use. For this purpose, the creators of DBLP provide an URL [5] to obtain the XML file that contains the necessary data sets. However, the file's current size is about 1.2 GB, so we strongly advise to use a computer with adequate computational performance to parse the file. A disadvantage of this method is missing out on updates that are made to the online database. This makes it necessary to re-download it in periodic intervals.

For those reasons, we decided to make use of the third alternative for our project: the DBLP-provided web API, which delivers requested data in several formats. Comprehensive documentation by the creator of DBLP is provided in [6].

2) Querying information - an example

Our application will access DBLP data records by retrieving author information, parsing it and then requesting additional data records based on the processed data. The following example will illustrate how to retrieve data records for the author name "Michael Ley".

First of all, the author name has to be traced back to a unique identifier which describes this author in the database. DBLP has XML records for many of their HTML pages. The following URL will return an HTML page. However with a small modification we can change the output into XML.

- HTML
  http://dblp.uni-trier.de/search/author?
  author=last_name:first_name

- XML
  http://dblp.uni-trier.de/search/author?
  xauthor=last_name:first_name

The above http-request triggers a search for the author in the DBLP database and returns an XML with author names matching or similar to the search request parameter. The URL for accessing the author record of "Micheal Ley" is as follows:

- http://dblp.uni-trier.de/search/author?xauthor=ley:michael

The corresponding XML result is below:

```
<authors>
  <author urlpt="l/Ley:Michael">Michael Ley</author>
  <author urlpt="l/Leyer:Michael">Michael Leyer</author>
  <author urlpt="l/Leyton:Michael">Michael Leyton</author>
</authors>
```

Figure 1 - Query result for name search

According to the output above, there are three authors with a matching name in the database. However, only the first record fits our search criteria. Now we can obtain the author's identifier by simply copying the value of the "urlpt" attribute. From there we can retrieve several kinds of related information. We can now proceed with retrieving the author's data record by performing the following query:

- http://dblp.uni-trier.de/pers/xx/**l/Ley:Michael**

Due to the enormous size of the result, an excerpt of the output can be found in the appendix [A].

3) Limitations

Although an API is provided for retrieving information from the DBLP database, complex queries or even joins cannot be performed through it. As mentioned earlier, this is due to the fact that the database is run with helper utilities instead of a DBMS. Thus, the logical linking of data has to be done on the client application. Furthermore, not all data records which are of interest to this project can be retrieved by using the API. This makes a search for certain fields (e.g. Journal title) impossible.

Of course, downloading the database and parsing it locally would remove those limitations. However, we do not possess the required computing capacity for this and also want avoid the necessity of periodically updating the local database copy. However not having a copy of the records locally, has its advantages and disadvantages. On the bright side, we didn't have to build a DBMS system and therefore there is no maintenance cost to it. Bad part of this is we have to rely on stability of other systems. If they decided to change their structure we had to modify our implementation accordingly.
Because of these limitations, our application's main functionality revolves around author search.

B. Mission Statement

The goal of this paper is the development of an application that retrieves data from DBLP, an online bibliography database for computer science research papers, and linking them with information from various other online resources, such as DBpedia and IEEE. Furthermore, statistics should be generated from the DBLP results.
Our approach to the result set is not to create a timeline of publications, but we focused on grouping similar items together and presenting more compact display to users where they can reach many features, which will be explained in the next section, from a single link from our implementation.

C. Functionality

The application shall provide the following features:

- Perform search based on author name
- Retrieval and display of data sets from DBLP database
- Find co-authors
- Link data fields of a search result DBpedia, IEEE and parse them
- Create timeline for a specific conference and map locations to Google Maps
- Display statistics for a certain type of information, e.g. publications per author, frequent collaborators

II. Related Work

There are several works that have been using the DBLP and DBpedia services.
For example, [13] proposed a framework architecture supporting the development of more complex mashups incorporating dynamic data integration. The framework consists of components for query generation and online matching as well as for additional data transformation. The framework architecture supports interactive and sequential result refinement to improve the quality of the presented result by executing more elaborate queries when necessary. A script-

based definition of mashups facilitates the development as well as the dynamic execution of mashups.

They illustrate their approach by a powerful mashup implementation combining bibliographic data to dynamically calculate citation counts for venues and authors. The framework supports a script-based definition of mashups and the use of multiple query strategies for accessing external data sources. Query results can be dynamically refined to reach a good trade-off between fast execution times and high result quality.

Another related work SemWeB [14], it's an extension to the Mozilla Firefox Web browser. SemWeB adds a semantic layer to Web Documents; it annotates Web pages using a linked data domain (i.e. DBpedia) and creates context-based hyperlinks on Web pages to guide users to relevant pages. In addition, the information presented to the user is personalized based on a novel behavior based user model. They evaluated their approach on DBpedia, DBLP and ECS linked data domains. SemWeB provides a new way of supporting dynamic linking and personalization on Web documents using different linked data domains.

Their focus was on the benefits of using linked data for annotating arbitrary Web resources, generating context-based hyperlinks and providing personalization in open corpus Web with a novel Semantic Web browser, called SemWeB.

SemWeB utilizes linked data for understanding the context of a Web page, and then it creates and embeds context-based hyperlinks to the page; In addition, they developed a novel user model for Web personalization and integrated this architecture to the SemWeB.

Other related work the development of DBLP-SSE[15] (A DBLP Search Support Engine) which is mainly provide two types of supporting functionalities, namely, search refinement support and domain analysis support. For the search refinement support, DBLP-SSE first track the change of each author research interests and make a prediction of his/her current research interests based on some interest retention models, then it use acquired user interests as implicit query constraints to refine incomplete or vague queries from users. Through this supporting functionality, search results which are consistent with predicted user interests are ranked into the top ones, and users can easily find the results which may be most relevant to their needs although they may not explicitly put enough necessary constraints to the input query. For the domain analysis support, DBLP-SSE provides support on building domain structures, tracking domain trend, finding author distributions, etc. These functionalities are also user centered, if the users log on the system, based on predicted user interests, the system can automatically generating relevant domain analysis results to users so that they can be aware of the change in this domain.

III. Implementation

A. General Concept

For our implementation we focused on reaching all possible information dynamically by using either XML results or parsing HTML content. We do not store any of the information locally in a file or on a database.

Main function of our implementation is to retrieve keys from DBLP XML results, and forming new XML queries to retrieve more information about an author or URLs to retrieve more information about conferences as explained in Section 1.A.2

We used Amazon Web Services [11] to deploy our web site. Our AWS server[1] is running on a 64 bit Ubuntu 12.04 LTS, kernel 3.2.0 provided by AWS free-tier subscription. Along with that, in order to publish our web site, we used Apache2 (version 2.2.22) web server, and PHP5 (version 5.3.10) for development, as well as pChart 2.0 [9] library with GNU GPLv3 license to draw the graphs, and Google Maps API [10].

B. Implementation of Functionalities

For the functionalities listed on Section 1.C we queried DBLP servers with format explained in Section 1.A.2 in order to retrieve the XML version of the result. Once we got the XML results, we used PHP - The DOMDocument class [12] in order to parse the result and get all listed "items" for that author. The item types in DBLP are;

- Books and Theses
- Journal Articles
- Conference and Workshop Papers
- Parts in Books or Collections
- Editorship
- Reference Works
- Informal and Other Publications

From the above items, most common types are *Conference and Workshop Papers*, *Journal Articles*, *Books and Theses*, *and Editorship*. For that reason our implementation mainly focused on these four types, since finding examples for all is hard.

By forming the query with key names for a specific author, our implementation can get the following results;

- Journal articles list,
- Books,
- Editorship,
- Conference and Workshop papers,
- Co-author list,
- Statistical graphs based on above retrieved results,
- Google Maps results for conference history of the author.

[1] http://ec2-184-73-151-51.compute-1.amazonaws.com/DBLP/dblp.php

All of the above results can be obtained by parsing a XML result along with some extensions. Some of the extensions we added are parsing HTML pages for mapping conferences on a Google Map, re-querying the DBLP server, for more result in order to create the author graph and linking ISBN numbers to an external website[2] in order to provide more information about a book through an external link.

Moreover, if DBLP has a link to databases that holds more information, we provide external links to those databases like ACM, IEEE, JSTOR, and Science Direct from Journal Articles or Conference papers. Also, we provide single XML records for each entry and link to DBLP server for bibliographic information as shown in Figure 2.

```
@proceedings{DBLP:conf/gvd/2007,
  editor    = {Hagen H{\"o}pfner and
               Friederike Klan},
  title     = {Post-Proceedings of the 19. GI-
               Workshop on Foundations of
               Databases (Grundlagen von
               Datenbanken), Bretten,
               Baden-W{\"u}rttemberg,
               Germany, May 29 - June 1, 2007},
  booktitle = {Grundlagen von Datenbanken},
  publisher = {School of Information Technology,
               International University
               in Germany},
  series    = {Technical Report},
  volume    = {02/2007},
  year      = {2007},
  bibsource = {DBLP, http://dblp.uni-trier.de}
}
```

Figure 2 - DBLP bibliographic information sample

C. Conference History with Google Maps API

The goal of this feature is to visualize the various locations that a conference has taken place in. For this purpose, we make use of the Google Maps API. Google provides an excellent example on how to invoke the API with AJAX [7]. The code presented in the aforementioned example is embedded into our application with a slight modification: instead of retrieving the markers from a MySQL database, they are generated with a PHP script. This script takes the key of a given publication and retrieves the short abbreviation of the conference associated with it.

A publication is stored in an author record and has the following format:

```
<r>
  <inproceedings key="conf/gvd/Ley07" mdate="2010-10-25">
    <author>Michael Ley</author>
    <title>Datenqualität: Eine organisatorische und technische
    Herausforderung - Erfahrungen von der DBLP-
    Bibliographie.</title>
    <pages>1</pages>
    <year>2007</year>
    <crossref>conf/gvd/2007</crossref>
    <booktitle>Grundlagen von Datenbanken</booktitle>
    <url>db/conf/gvd/gvd2007.html#Ley07</url>
  </inproceedings>
</r>
```

Figure 3 - Sample of a publication on XML result

The key attribute contains the identifier of the publication. By truncating the string at the last occurrence of the slash (‘/’), the abbreviation of the conference can be obtained. Then information about the history of the conference can be retrieved by parsing the HTML result of the following URL:

- http://dblp.uni-trier.de/**db/conf/gvd/**

Unfortunately, information about the conference cannot be requested as an XML document. However, we only need title, location and annual number of the conference in order to create markers on the Google map. This information is stored in the <h2> elements of the HTML response and can be parsed easily.

```
<h2>25. GvD 2013: Ilmenau, Germany</h2>
<h2>24. GvD 2012: Lübbenau, Germany</h2>
<h2>23. GvD 2011: Obergurgl, Austria</h2>
…
```

Figure 4 - HTML Response for Conference History

The parsed information is converted into an XML document in order to create the map and corresponding markers with the Google Maps API. The converted version is as follows;

```
<markers>
  ...
  <marker
    number="24"
    name="23. GvD 2011" address="Obergurgl, Austria"
    lat="46.8701600" lng="11.0273000" type="Conference"/>
  <marker number="25"
    name="24. GvD 2012" address="Lübbenau, Germany"
    lat="51.8632976" lng="13.9545465" type="Conference"/>
  <marker number="26"
    name="25. GvD 2013" address="Ilmenau, Germany"
    lat="50.6843502" lng="10.9254728" type="Conference"/>
</markers>
```

Figure 5 - XML Result of parsed Conference History

D. Statistical Graphs with pChart Framework

Goal of this module is to create some visual representation of the data belonging to queried author. Since we parsed the data from DBLP, we basically have freedom to create any form of visual representation. The only requirement to create a graph is to have the data ready in an array. We decided to implement 7 different graphs with the data we have about the author and they can be listed as;

1. Total number of publications per year,
2. Total counts of different published items,

[2] http://www.isbnsearch.org/

3. Total number of each other authors contribution to the author,
4. Percentage of published items with each other,
5. Relation of author with co-authors
6. Frequency of per attended conferences,
7. Comparison of published Journals vs. Conferences per year.

The most challenging graph among these 7 graphs was the author relations graph. Since we do not store any data locally, we had to open up a new connection for each co-author and another connection for co-author's co-author list. Basically, we went 2 levels deep on our relation graph.

E. Challenges and Limitations

The main challenge we have with our implementation is that it, as we explained earlier, completely relies on results obtained from the DBLP server. Since data stored in the DBLP server is also entered manually, some errors might occur or some of the data might not appear. For example, not all the items have the most common tags as all other similar items have. Some journals might not have a number or issue tag. These situations are not very common, therefore those differences has to be fixed by locating the error and manually fixing the code to satisfy the change.

Another challenge we faced with graphs is that our author graph takes too much time to process, since it needs to open several connections to the DBLP server in order to complete populating the required co-author lists. Since this processing might take more than 30 seconds, we had to limit our co-author number with maximum of 10 authors on both levels.

The main limitation we have is, since we do not store any data locally, we need to be able to reach the data by forming an URL and parsing the results either from XML or HTML. As long as the server we need to query has a standard URL rules in terms of reaching an object (e.g. item, author, conference, location, etc.) we can form an URL and easily parse the results obtained.

IV. Conclusion and Future Work

Our aim in this project is to use data provided mainly by DBLP and convert it based on our needs and come up with our version of a dashboard which will give more accessible visual representation of data to user and along with that, provide more external data obtained from several other sources.

This project has no implementation limits, since data on the web continue to emerge and more and more sources will be available to users every other day. The possibilities of adding more information, different representation of same data and reaching out several more resources are diverse, and we will continue add more sources to this dashboard.

Our additions to original DBLP project are;

- Easily accessible, easy to use dashboard
- Quick summary of queried author,
- Graphical representation of statistical data for queried authors,
- Map representation of conferences' timeline,

A recommendation for future work is connecting DBpedia triplets to objects, such as journals, conferences or authors in order to provide more information about them. Additionally, adding citation information for each specific publication to help researchers find more related publications should be considered.

References

[1] Michael Ley. 2002. The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. *In Proceedings of the 9th International Symposium on String Processing and Information Retrieval* (SPIRE 2002), 1-10.

[2] Michael Ley. 2009. DBLP: some lessons learned. *Proc. VLDB Endow*, 1493-1500.

[3] DBLP - How does the 'author search' work? Retrieved November 15, 2013, from http://www.informatik.uni-trier.de/~ley/db/about/author.html

[4] Faceted DBLP, Retrieved November 15, 2013, from http://dblp.l3s.de/

[5] DBLP - XML distribution folder. http://dblp.uni-trier.de/xml/

[6] Michael Ley. 2009. DBLP XML Requests. http://dblp.uni-trier.de/xml/docu/dblpxmlreq.pdf

[7] Ben Appleton. 2007. Using PHP/MySQL with Google Maps. Retrieved November 15, 2013, from https://developers.google.com/maps/articles/phpsqlajax_v3

[8] Complete Search DBLP. Retrieved November 15, 2013, from http://www.dblp.org/search/

[9] pChart PHP library. http://www.pchart.net/

[10] Google Maps API. https://developers.google.com/maps/

[11] Amazon Web Services Homepage. http://aws.amazon.com/

[12] PHP DOMDocument class. http://php.net/manual/en/class.domdocument.php

[13] Rahm, Andreas Thor David Aumueller Erhard. 2007. Data integration support for mashups. *Workshops at the Twenty-Second AAAI Conference on Artificial Intelligence* (AAAI-07)

[14] Melike Şah, Wendy Hall, and David C. De Roure. 2010. Dynamic linking and personalization on web. *In Proceedings of the 2010 ACM Symposium on Applied Computing* (SAC '10). pp. 1404-1410. doi:10.1145/1774088.1774386

[15] Yi Zeng; Yiyu Yao; Ning Zhong, "DBLP-SSE: A DBLP Search Support Engine," *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on*, vol.1, no., pp.626-630, doi:10.1109/WI-IAT.2009.364

Appendix A

```
<dblpperson name="Michael Ley" n="31">
  <person key="homepages/l/MichaelLey" mdate="2012-10-29">
    <author>Michael Ley</author>
    <url>http://www.informatik.uni-trier.de/~ley/</url>
    <url>http://www.orcid.org/0000-0001-7580-4351</url>
    <url>http://scholar.google.com/citations?user=2jE4KhkAAAAJ</url>
    <url>http://academic.research.microsoft.com/Author/596085/michael-ley</url>
    <url>http://dl.acm.org/author_page.cfm?id=81100182162</url>
  </person>
  <r>
    <article key="journals/pvldb/Ley09" mdate="2009-09-02">
      <author>Michael Ley</author>
      <title>DBLP - Some Lessons Learned.</title>
      <pages>1493-1500</pages><year>2009</year>
```

```
        <volume>2</volume>
        <journal>PVLDB</journal>
        <number>2</number>
        <ee>http://www.vldb.org/pvldb/2/vldb09-98.pdf</ee>
        <url>db/journals/pvldb/pvldb2.html#Ley09</url>
      </article>
    </r>
    <r>
      <inproceedings key="conf/gvd/Ley07" mdate="2010-10-25">
        <author>Michael Ley</author>
        <title>Datenqualität: Eine organisatorische und technische
Herausforderung - Erfahrungen von der DBLP-Bibliographie.</title>
        <pages>1</pages>
        <year>2007</year>
        <crossref>conf/gvd/2007</crossref>
        <booktitle>Grundlagen von Datenbanken</booktitle>
        <url>db/conf/gvd/gvd2007.html#Ley07</url>
      </inproceedings>
    </r>
    ...
</dblpperson>
```

APPENDIX B

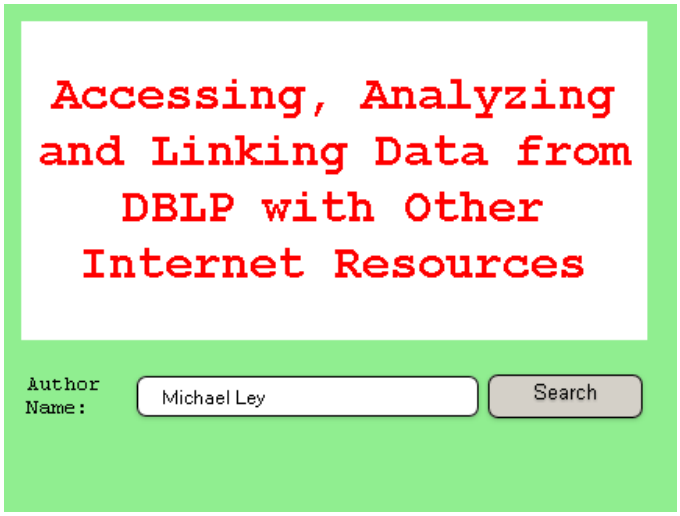

Figure 6 - Search Page

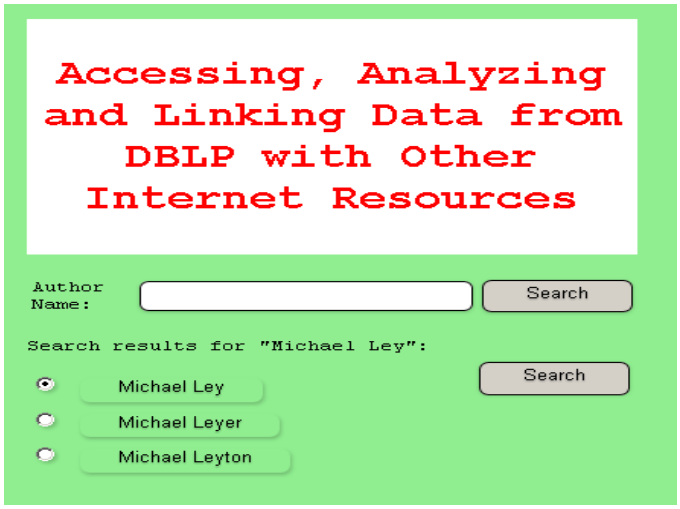

Figure 7 - Search Results

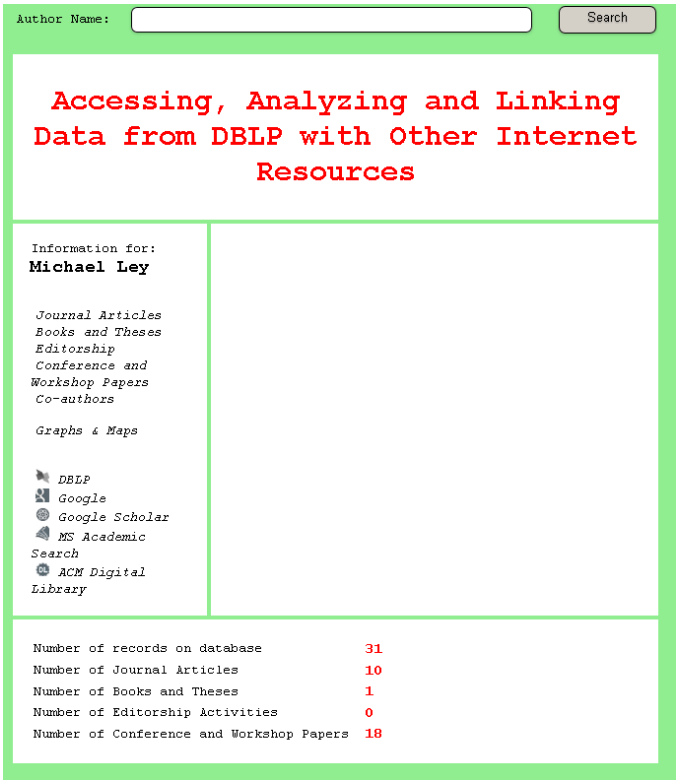

Figure 8 - Author Homepage

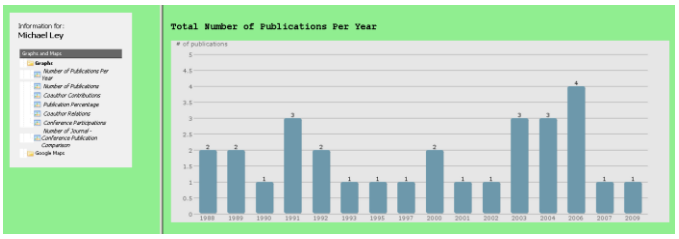

Figure 9 - Sample Graph Page

Figure 10 - Sample Google Maps Page