A data extractor system for extraction, deformatting, and postformatting of data available on the WWW. A user, connected to the Internet/Web, contacts a fee-based intermediate data service which provides an interface to determine various aspects of the user's query, including output format. The intermediate data service generates a Java stripping agent which is sent to the user's browser to interface with a third party data provider. The Java stripping agent contains the knowledge to strip away the formatting of user interfaces. The system will allow buffering and streamlining between the user and web data providers; converting visual presentation of information into data for further processing, translating one data request into a cascade of data requests and pasting results together, filtering data output; allowing a variety of presentations of data different from the original presentation; optional dataflow between the user's applications and the third-party data providers bypassing interactive interfaces.

32 Claims, 9 Drawing Sheets
FIG. 1
With this application you may get enhanced data from various Information Services Providers.

You can save results of your query as various data formats.

As example, you can see the result of query to "Sears" via BellSouth:

- Text (tab-delimited) format
- HTML format
- Excel format
- Microsoft Access format

FIG. 2
ENHANCED INFORMATION SERVICES

Enhanced information by Search\textsuperscript{(TM)}

Keyword(s):

Output will be in: Plain text(tab delimited)

Output language: English

We also enhance other information services: LawStreet

(you can bookmark the result)

FIG. 3
<table>
<thead>
<tr>
<th>Block H &amp; R-Sears Location</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sears Car Rental</td>
<td>3905 Newton Rd</td>
<td>Albany</td>
<td>GA</td>
<td>434-4300</td>
</tr>
<tr>
<td>Sears Portrait Studio</td>
<td></td>
<td>Albany</td>
<td>GA</td>
<td>434-4351</td>
</tr>
<tr>
<td>Sears Products &amp; Services</td>
<td>2815 Gillionville Rd</td>
<td>Albany</td>
<td>GA</td>
<td>483-9898</td>
</tr>
<tr>
<td>Sears Roebuck &amp; Co-General Information</td>
<td>2601 Dawson Rd</td>
<td>Albany</td>
<td>GA</td>
<td>434-4300</td>
</tr>
<tr>
<td>Sears Roebuck &amp; Co-General Information</td>
<td>1408 E Shotwell St</td>
<td>Bainbridge</td>
<td>GA</td>
<td>246-5877</td>
</tr>
<tr>
<td>Block H &amp; R-Sears Location</td>
<td>3401 Masonic Dr</td>
<td>Alexandria</td>
<td>LA</td>
<td>71301(318)</td>
</tr>
<tr>
<td>Sears Authorized Retail Dealer</td>
<td>636 Moreau St</td>
<td>Bunkie</td>
<td>LA</td>
<td>71322(318)</td>
</tr>
<tr>
<td>Sears Authorized Retail Dealer</td>
<td>363 Moreau St</td>
<td>Marksville</td>
<td>LA</td>
<td>71351(318)</td>
</tr>
<tr>
<td>Sears Optical</td>
<td>3401 Masonic Dr</td>
<td>Alexandria</td>
<td>LA</td>
<td>71301(318)</td>
</tr>
<tr>
<td>Sears Portrait Studio</td>
<td>3401 Masonic Dr</td>
<td>Alexandria</td>
<td>LA</td>
<td>484-7251</td>
</tr>
<tr>
<td>Sears Roebuck &amp; Co</td>
<td>3401 Masonic Dr</td>
<td>Alexandria</td>
<td>LA</td>
<td>484-7200</td>
</tr>
<tr>
<td>Sears Roebuck &amp; Co-General Information</td>
<td>900 N 5th St</td>
<td>Leesville</td>
<td>LA</td>
<td>71446(318)</td>
</tr>
<tr>
<td>Sears Hearing Aid Center</td>
<td>3101 N Main St</td>
<td>Anderson</td>
<td>SC</td>
<td>29621(864)</td>
</tr>
<tr>
<td>Sears Portrait Studio</td>
<td>3131 N Main St</td>
<td>Anderson</td>
<td>SC</td>
<td>29621(864)</td>
</tr>
<tr>
<td>Sears Roebuck &amp; Co-Clemson-Authorized Retail Dealer Store-General Information</td>
<td>1019 Tiger Blvd</td>
<td>Clemson</td>
<td>SC</td>
<td>29631(864)</td>
</tr>
<tr>
<td>Sears Steve Garage</td>
<td>186 Cherokee Rd</td>
<td>Williamston</td>
<td>SC</td>
<td>29697(864)</td>
</tr>
</tbody>
</table>

**FIG. 5**
The SQL queries to BellSouth Yellow Pages Database

You can issue one or many SQL queries to BellSouth Yellow Pages.

- The name of the table, which consists Yellow Pages is AllBell.
- Each SQL query should have the semicolon (;) marker on the end.
- The results of the last query could be displayed on the screen.
- You should specify the list of field names for getting thes results.
- The number of output lines is limited to 500 lines.

Please, look into examples of SQL queries and feel free to modify them or put your own SQL query:

- Business, which are located in some zip-code
- Count all Business, which are located in some zip-code (Count example)
- Select phones like ‘348%’ (results is not displayed - just for demonstration how to issue many SQL commands) then Show all distinct cities.

Output columns:

Businessname HouseNumber Street City State ZIP AreaCode Phone

Series of SQL statements (please, end with ; symbol each of these statements):

Tab-delimited text

FIG. 6
The SQL queries to BellSouth Yellow Pages Database:

Output columns:

Businessname HouseNumber Street City State ZIP AreaCode Phon

Series of SQL statements (please, end with ; symbol each of these statements):

Select * from AllBell where zip=33199;

FIG. 7A
The SQL queries to BellSouth Yellow Pages Database:

Output columns:
- CountResults

Series of SQL statements (please, end with ; symbol each of these statements):
```
Select count (*) as CountResults from AllBell where zip=33174
```

FIG. 7B
The SQL queries to BellSouth Yellow Pages Database:

Output columns:
City

Series of SQL statements (please, end with ; symbol each of these statements):
Select * from AllBell where Phone Like '348%';
Select distinct city from AllBell;

FIG. 7C
1. Field of Invention

The present invention relates generally to the field of data retrieval. More specifically, the present invention is related to a business model which provides a fee-based, real-time, intermediary service including a method of extracting data from third party providers, removing existing formatting information and returning the data to the requester in a desired format.

2. Discussion of Prior Art

The proliferation of the Internet and World Wide Web (WWW) has produced a deluge of information often times in unmanageable formats to the average user. To assist the user, various search engines have been developed which work through the user's browser to keyword search various indexed data sources. While search results of text Web pages may be easy to manage, search results of structured type data prove not to be so easily managed. Typically database results are returned preformatted in HTML, text or spreadsheet forms. The user, however, has no means of selecting a format not envisioned by the data supplier. The user may want to select a data output only in spreadsheet format for direct integration into locally stored table structures. Most users cannot perform such a conversion because of software or hardware limitations, and certainly not in real-time. What is needed is an intermediate service provider through which a user can enhance their data retrieval by customizing the data output without having to create complex algorithms or mapping structures locally on their PC. The following prior art describes various attempts to extract data from database sources located on the Web.

The patent to Schofield (U.S. Pat. No. 5,860,072), assigned to Tandem Computers Incorporated, provides for a Method and Apparatus for Transporting Interface Definition Language-Defined Data Structures Between Heterogeneous Systems. Data strings are stored locally in a receiving computer's buffer and thereafter, the data structure extracted, realigned and stored. Column 4, lines 37–39 suggest an Internet embodiment.

The patent to Horvitz et al. (U.S. Pat. No. 5,864,848), assigned to Microsoft Corporation, provides for a Goal-Driven Information Interpretation and Extraction System. Column 1, lines 47–52 suggest the extraction of data from Internet web pages.

The page entitled, “Visual Design and Cross-Platform Development”, provides for a technical overview of the software product “Cambio.” Cambio extracts the desired data fields (which can be spread across multiple lines in a text file) and assembles those fields into a flat record of data. These records are presented in the conventional row/column, tabular format (see http://www.datajunction.com/products/cambio_technical.html).

The web page entitled, “GlimpseGate”, provides for context searching of html web documents with data strings (see http://phones.cybercell.net/~hfs/sources/glimpsegate/).

Additional data extractors can be found in the following patents, web pages and articles:


Web sites:

4. Overview -http://skwww.unc.edu/iis/node42.html;

SUMMARY OF THE INVENTION

A data extractor system for the extraction, deformatting, and postformatting of data available on the WWW. The system enables buffering and streamlining between the user and web data providers; converting the visual presentation of information into data for further processing, translating one data request into a cascade of data requests and pasting results together, filtering data output; allowing a variety of presentations of data different from the original presentation; optional dataflow between the user’s applications and the third-party data providers thereby bypassing interactive interfaces.

A user, connected to the Internet/Web, contacts an intermediate data service which provides an interface to determine various aspects of the user’s query, including output format. The intermediate data service generates a stripping agent, such as a Java program, which is sent to the user's browser to interface with a third party data provider. The Java stripping agent contains the knowledge to strip away the formatting of user interfaces such as HTML, reformat, reorganize, filter and present the data in real-time in a user-selected format. The present invention:

1. Embeds all user input in a standardized way in a URL (CGI), hiding from the user various data entry protocols such as post-data, Java script data entry forms, etc. Thus, allowing the user to:
   a. bookmark this URL with predefined input data
   b. embed this URL in various user scripts

2. Converts the formatted data retrieved from third party data provider into an ASCII file, one line per result, tabs separating fields; eliminating all graphics and irrelevant text, leaving only data allowing:
   a. convenient downloading of data into user applications
   b. compact results
   c. development of embedded applications
3. When a third-party site gives a few records at a time and a "next" button, the present invention recursively dispatches an agent to recursively call the third-party data provider to give the user in one operation a large volume of data.

4. In addition to plain ASCII output by default, the user will be able to parametrically specify additional forms of output:
   - formatted ASCII (72 characters per line, aligned spaces instead of tabs, one field can continue on several lines)
   - RTF
   - HTML tables
   - PDF
   - Postscript
   - And others

The present invention delivers standardized extracted graphic files of spatial data: maps, remote-sensing images, etc.

5. The user can specify a parameter EGREP_SCREEN giving a regular expression to screen the output or a simplified parameter KEYWORDS_SCREEN. (Note: this is post-processing of results after they are received from third-party providers)

6. In an alternative embodiment, the intermediate data service subscribes to a variety of pay-per-use services and re-delivers information to paying customers. The end user's convenience, in addition to repackaging, will be that the user does not have to subscribe to many services, just to the intermediate data service (a charge includes a small mark-up, or no mark-up if wholesale rates are obtained).

7. In an alternative embodiment, the system performs merges and joins between data from more than one server.

8. In an alternative embodiment, certain joins will be allowed within the same site, e.g., by traversing pointers to product detail from product list.

9. In an alternative embodiment, the system includes a virtual conceptual semantic schema of all WWW information accessible by the user via the service and allow the user to specify complex database query against the schema without knowing which third-party sites need to be accessed or joined to perform the query.

10. The program can employ Java-agent technology, which agent will perform all the activities at user site; reducing traffic on the intermediate data service and will also protect the intermediate service provider from possible claims of third-party data providers regarding reselling or storing of their data contrary to license or copyright provisions.

11. The program will allow a number of post-formatting options, including:
    - audio file produced after adding connecting words to properly delineate fields (it is impossible to produce a meaningful audio file without first stripping output and delimiting fields with connecting words)
    - smart translation into other languages. The present invention will decide which fields should be translated and which should not, exercising its knowledge of the semantics of the data source.

12. The program is written in such a way that definitions of the third-party web site protocols are outside of the program, in a Knowledge Base, and easy to maintain and change by a low-skilled staff.

13. The intermediate data service maintains a large database or references to data providing sites whose input/output stripping instructions are known.

14. When no parameters are given, the present invention replies with a list of third party services it knows to query, the kind of information they provide, and list of field names.

15. Examples of services to be supported are:
    - various white and yellow phone directories
    - business directories and classification (SIC)-zip2.com weather services
    - stock quotes (input: a list of ticker symbols)
    - public English dictionaries, bilingual dictionaries, and thesauri
    - web search engines (Dog Metaland; Yahoo; Infoseek)
    - geographic text servers (zipcodes -> city, address -> area code -> airport code)
    - online translators
    - airline schedules and flight info (airline-specific sites)
    - professional directories: doctors, lawyers
    - Microsoft aerial photography maps

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a flowchart of the present invention.

FIG. 2 illustrates an enhanced information services interface.

FIG. 3 illustrates an enhanced keyword search interface.

FIG. 4 illustrates an example output in text/tab-delimited format.

FIG. 5 illustrates an example output in HTML format.

FIG. 6 illustrates an SQL interface.

FIGS. 7a–c collectively illustrate a SQL example.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

While this invention is illustrated and described in a preferred embodiment, the device may be produced in many different configurations, forms and materials. There is depicted in the drawings, and will herein be described in detail, a preferred embodiment of the invention, with the understanding that the present disclosure is to be considered as a exemplification of the principles of the invention and the associated functional specifications of the materials for its construction and is not intended to limit the invention to the embodiment illustrated. Those skilled in the art will envision many other possible variations within the scope of the present invention.

FIG. 1 illustrates a flow diagram of a user 102 connecting to a data supplier 104 to perform a search during a typical search session using known Internet/WWW search engines such as Lycos®, Excite®, Snap®, Infoseek®, Webcrawler®, etc. User 102 represents a PC owner with Internet access and a browser 110 (e.g., Netscape® or Microsoft Explorer®), WebTV®, or other Internet/WWW access methods.

The present invention provides for an intermediate data service enhancement 106 enabling the user to: strip 116 away the formatting of user interfaces such as HTML used by either the data provider 104 or browser 110, reformat, reorganize, filter and present the data 126 in a user selected format. User 102 connects 108 to intermediate data service enhancement 106 through their browser 110. Intermediate data service enhancement 106 provides a user with a search enhancement interface (FIGS. 2–7c) to determine a choice of data supplier 104, return data format, and query. Intermediate data service enhancement 106 returns 109 an javStrip class algorithm 116 to the user's system to enable
realtime local enhancement. The strip algorithm 116 retrieves the requested data 118/ 120 strips the non-data, formatting, reformats, reorganizes, filters and presents the data 126 in a user selected format 126.

FIG. 2 illustrates a typical user interface 200 provided by the intermediate data service enhancement 106. User 102 first selects an Information Service 202 such as LawStreet® (shown), BellSouth®, Excite®, Webcrawler®, Lycos®, Snap®, Goto®, Scrubtheweb®, MSN®, or a generic search engine—search and actuates this selection by selecting “Go”. Instructions provided include:

a. With this application you may get enhanced data from various Information Service Provider—206

b. You can save the results of your query as various formats—208. You can make advanced ad-hoc queries 210

c. Working example(s)—212 in formats 214 (text (tab-delimited), HTML, Excel®, Microsoft Access®).

Upon actuation of the “Go” button 204, the user receives the next user interface 300 as shown in FIG. 3. Entry box 302 enables the user to enter typical keyword(s) normally used during a search. Drop-down menu 304 enables the user to select a desired output format such as “plain text (tab-delimited)”—shown, HTML, Excel®, Microsoft Access®, or other known data formats. Drop-down menu 306 enables a selection of language, e.g., English (shown), French, German, Italian, etc., for the returned data. Upon selecting keywords, output format and language, the user can start the creation of a java strip algorithm by selecting “Go” or change information service providers at 202. In addition, the user can bookmark 310 the result for future access. Intermediate data service enhancement 106 returns java strip class algorithm (Java strip agent) which works locally with the user’s browser 110 to return a “data only” result in the format selected by the user selections registered in the steps outlined above. All Java® strip agents 116 created by the intermediate data service enhancement 106 system are retained therein for quick future access by a requesting user.

FIGS. 4 and 5 illustrate two possible outputs of the example 212. In this example, the user selected “BellSouth” as the service provider 202, “Sears” as the keyword 302, “English” as the output language 306 and “plain text (tab-delimited)” as the output shown in FIG. 4 and HTML as shown in FIG. 5.

In addition to simple keyword searches, advanced ad-hoc queries 210 can be made with relative ease. FIG. 6 illustrates the SQL (structured query logic) query interface 600 with BellSouth Yellow Pages chosen as the service provider. The user is given guidance instructions 602–622 describing basic SQL procedures related to BellSouth databases as follows:

**BASIC PARAMETERS OF THE DATABASE**

The user can issue one or many SQL queries to BellSouth Yellow Pages. 602

The name of the table, which consists Yellow Pages is “AllBell”—604

Each SQL query should have the semicolon(;) marker on the end—606

The results of the last query could be displayed on the screen—608

The user should specify the list of field names for getting these results—610

The number of output lines is limited to 500 lines.—612

**EXAMPLES**

Please look into examples of SQL queries and feel free to modify them or put your own SQL query:—614

Businesses which are located in some zip-code—616

Count all businesses, which are located in some zip-code (count example)—618

Select phone like “348%” 620, then show all distinct cities—622

As each hypertext example 616–622 is selected, the interfaces shown in FIGS. 7a–7c reveal the actual SQL query entered into the series of SQL statements entry box 626. FIG. 7a illustrates the SQL entry 701 “Select * from AllBell where zip=33199”; 716 correlating to the text example 616. FIG. 7b illustrates the example 618 which produces the SQL entry 702 “Select count(*) CountResults from AllBell where zip=33174”; 718. FIG. 7c illustrates the examples 620/622 which produce the SQL entries 703 “Select* from AllBell where Phone Like 348%”; 720; Select distinct city from AllBell;—722.

The remainder of SQL selections include Output columns 624 desired for data output and output format 628. “Go” 628 actuates the SQL process, creating the Java® Strip Class algorithm 116 equating to the SQL and enhanced data output selections which is then returned to the user 102.

The dispatched agent retrieves and reformats the data. Limited amounts of results are delivered to the user at no charge. Larger amounts are charged in correlation to the amount of data retrieved. The user has the option to order a preview and sampling of data before the full set is delivered and the charge is effected. In an alternative embodiment, the user purchases a license for unlimited utilization of the service. Additionally, the service can be provided free to the user and payments made by advertisers or other third parties. Furthermore, in some situations, no charges would be incurred at all.

Format of the default ASCII output follows:

row separator: newline (+ optionally carriage return if parameter DOS=y is given)

field separator: tab or other user-specified delimiter structure:

\<document titles>

\<lines of column headers>

==

\<data>

====

\<informational messages, including sites contacted queries performed, time stamps, statistics>

\<error messages>

=====

\<optional promotional material and paid advertisements>

\<links to third party services used>

The above enhancements for data extraction and its described functional elements are implemented in various computing environments. For example, the present invention may be implemented on a conventional IBM PC or equivalent, multi-nodal system (e.g. LAN) or networking system (e.g. Internet, WWW). All programming, Java strip agent algorithms, GUIs, display panels and dialog box templates, and data related thereto are stored in computer memory, static or dynamic, and may be retrieved by the user of the present invention in any of: conventional computer storage, display (i.e. CRT) and/or hardcopy (i.e. printed) formats. The programming of the present invention may be implemented by one of skill in the art of database, Internet related and E-commerce programming.

**CONCLUSION**

A system and method has been shown in the above embodiments for the effective implementation of a data
extractor. While various preferred embodiments have been shown and described, it will be understood that there is no intent to limit the invention by such disclosure, but rather, it is intended to cover all modifications and alternate constructions falling within the spirit and scope of the invention as defined in the appended claims. For example, the present invention should not be limited by computer operating system, database management system, database management model, directory structure, DBMS-file linking technology, the type of user interface, computer hardware platform, network operating system, programming language of the agents, archiving software, or archiving hardware. In addition, the present invention can be implemented locally on a single PC, connected workstations (i.e. networked LAN), across extended networks such as the Internet or using portable equipment such as laptop computers or wireless equipment (RF, microwaves, infrared, photonic, etc.)

What is claimed is:

1. An E-commerce system generating revenues by providing a data extraction service for an end user comprising the method:
   receiving at said data extraction service a request from said end user comprising a database query to a data supplier;
   said data extraction service providing a graphical user interface to said end user, said graphical user interface requesting at least data output formatting requirements;
   said data extraction service generating a java agent based on at least said database query and formatting requirements;
   said data extraction service communicating said java agent to said end user;
   said end user implementing locally said java agent to extract data from said data supplier, said data further comprising multiple first data sets that are joined into a second data set, said second data set returned to said end user in said format specified, and said data extraction service billing said end user for use of said java agents.

2. An E-commerce system generating revenues by providing a data extraction service for an end user as per claim 1, wherein said data extraction includes stripping information and reformatting in the requested format.

3. An E-commerce system generating revenues by providing a data extraction service for an end user as per claim 2, wherein said step of stripping formatting information includes stripping HsML formatting.

4. An E-commerce system generating revenues by providing a data extraction service for an end user as per claim 2, wherein said step of stripping formatting information includes tripping graphics from HTML data.

5. An E-commerce system generating revenues by providing a data extraction service for an end user as per claim 2, wherein said stripping and reformatting is performed locally at the end user’s location.

6. An E-commerce system generating revenues by providing a data extraction service for an end user as per claim 5, wherein said stripping and reformatting is performed in real-time.

7. An E-commerce system generating revenues by providing a data extraction service for an end user as per claim 1, wherein said users are retained by said data extraction service for future use.

8. An E-commerce system generating revenues by providing a data extraction service for an end user as per claim 1, wherein said user interface further includes a selection from known data suppliers.

9. An E-commerce system generating revenues by providing a data extraction service for an end user as per claim 1, wherein said user interface further includes SQL query capability.

10. An E-commerce system generating revenues by providing a data extraction service for an end user as per claim 1, wherein said user interface further requests a language formal to return data to said end user.

11. A Web based system for data extraction comprising:
   - a data requestor connected to the Web;
   - a third party data provider;
   - an intermediate service provider, said intermediate service provider receiving a request for data from said requestor for data located at said third party data provider;
   - said intermediate service provider providing a graphical user interface to said requestor, said graphical user interface requesting at least data formatting requirements;
   - said intermediate service provider generating a java agent based on at least said request for data and formatting requirements;
   - said intermediate service provider communicating said java agent to said requestor;
   - said end user implementing said java agent to extract data from said data supplier, said data further comprising multiple first data sets that are joined into a second data set, and
   - said second data set returned to said end user in said format specified.

12. A Web based system for data extraction as per claim 1, wherein said requestor is billed for the services provided by said intermediate service provider.

13. A Web based system for data extraction as per claim 11, wherein said data extraction includes stripping formatting information and reformatting in the requested format.

14. A Web based system for data extraction as per claim 13, wherein said step of stripping formatting information includes stripping HTML formatting.

15. A Web based system for data extraction as per claim 13, wherein said stripping and reformatting is performed locally at the end user’s location.

16. A Web based system for data extraction as per claim 13, wherein said stripping and reformatting is performed in real-time.

17. A Web based system for data extraction as per claim 11, wherein said agents are retained by said intermediate service provider for future use.

18. A Web based system for data extraction as per claim 11, wherein said graphical user interface further includes a selection from known data suppliers.

19. A Web based system for data extraction as per claim 11, wherein said graphical user interface further includes SQL Capability.

20. A Web based system for data extraction as per claim 11, wherein said graphical user interface further requests a language parameter to return data to said end user.

21. A computer-based method of extracting data in a selected format from third party data providers comprising:
   - receiving from a request for data from a computer user connected to a computer network, contacting a data extraction service provider;
said data extraction service provider requesting at least a
source and output format of a data query, and
said data extraction service provider requesting to said
computer user a stripping agent to reformat said data
from said source format to said output format and to
enable said computer user to obtain said data from said
data source in the requested output format,
said data from said data source further comprising mul-
tiple first data sets and said reformatting of said data
further comprising joining said multiple first data sets
into a single, second data set.
23. A computer-based method of extracting data in a
selected format from third party data providers as per claim
22, wherein said stripping agent strips formatting informa-
tion including HTML.
24. A computer-based method of extracting data in a
selected format from third party data providers as per claim
22, wherein said stripping agent strips formatting informa-
tion including graphics from HTML data.
25. A computer-based method of extracting data in a
selected format from third party data providers as per claim
22, wherein said stripping and formatting is performed
locally at the computer user’s location.
26. A computer-based method of extracting data in a
selected format from third party data providers as per claim
22, wherein said stripping and formatting is performed in
real-time.

27. A computer-based method of extracting data in a
selected format from third party data providers as per claim
22, wherein said stripping agents are retained by said data
extraction service provider for future use.
28. A computer-based method of extracting data in a
selected format from third party data providers as per claim
22, wherein said step of said data extraction service provider
requesting at least a source and output format of a data query
is provided by a user interface.
29. A computer-based method of extracting data in a
selected format from third party data providers as per claim
28, wherein said user interface further includes SQL capa-
bility.
30. A computer-based method of extracting data in a
selected format from third party data providers as per claim
28, wherein said graphical user interface further requests a
language parameter to return data to said end user.
31. A computer-based method of extracting data in a
selected format from third party data providers as per claim
28, wherein said graphical user interface further includes as
selection from known data suppliers.
32. A computer-based method of extracting data in a
selected format from third party data providers as per claim
22, wherein said requestor is billed for the services provided
by said data extraction service provider.

* * * * *