Demonstration: A Robust Web Data-Extraction Technique With High Recall and Precision

D.M. Campbell, Y. Ding, D.W. Embley, K. Hewett, D.L. Jackman,
S.S. Jeffries, Y.S. Jiang, D. Lewis, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng,
A.L. Peacock, D.J. Seer, R.D. Smith, S.H. Yau, M. Xu, and L. Xu

Brigham Young University
Provo, Utah 84602, U.S.A.
Contact Author: liddle@byu.edu

Abstract

Our demo shows how to extract and structure data found in data-rich, unstructured, multiple-record Web documents. Users may either apply pre-built extraction applications or build and apply their own. The demo is significant because it (1) attacks an important data-centric problem and (2) uses database technology to produce good results with minimal effort.

Introduction

Over the past two years, we have experimented successfully with extracting data from data-rich, unstructured, multiple-record Web documents. Successful applications have included car ads and job ads [ECLS98], obituaries [ECN$^+$98,ECJ$^+$99], real estate, precious gems, computer monitors, games, musical instruments, stocks, and personals [Hom00].

We have made our extraction technology available on the Web [Hom00]. Both a "High-Level Data-Extraction Demo" (see Figure 1) and a "Detailed Data-Extraction Demo" (see Figure 2) are available. The two demos are the same, except that the detailed demo allows users to view intermediate steps and results.

Our approach to Web data extraction consists of the following five steps.

1. We begin with an HTML document that contains unstructured chunks of text for an application of interest. (The text box in Figure 3 shows a rendered HTML document for car ads – each car ad is an unstructured text chunk. The text box in Figure 4, which contains the HTML source for this document, represents the kind of input we process in our demo.)
2. For an application of interest we develop a conceptual-model instance, which we call an *application ontology*. An application ontology describes the application's objects and the relationships and cardinality constraints among objects. Each object set in the

ontology has a description of its lexical values and its context keywords, which aid in matching lexical constants identified in a document with object sets in the application ontology. (The text box in Figure 5 shows the first few lines of an application ontology for car advertisements.)

3. The system parses the application ontology to generate a database scheme and to generate matching rules for constants and keywords. (The "Database Scheme" box in Figure 6 shows the generated scheme for the cars application ontology.)
4. A record extractor automatically separates an unstructured Web document into individual record-size chunks, cleans the chunks of markup-language tags, and presents them as individual unstructured record text for further processing. ([EJN99] explains how the system accomplishes these tasks.)
5. A recognizer automatically applies the matching rules generated by the parser to the cleaned, unstructured records to extract data. The extraction algorithms use proximity heuristics to correlate extracted keywords with extracted constants and use cardinality constraints of the application ontology to construct records. The system places the extracted results in a database, which can then be queried using SQL. (Figure 6 shows a sample query and the results returned for the HTML source in Figures 3 and 4.)

To make our approach general and robust across new and changing Web pages, we fix in advance: the parser, the Web record extractor, the keyword recognizer, the constant recognizer, the database scheme generator, and the record data generator. To switch from one application domain to another, we simply switch to a different application ontology. We build a new one when we encounter a domain for which no application ontology exists.

To measure the success of our data-extraction work, we compute recall and precision ratios for each attribute for each application. We achieved recall ratios in the range of 90% and precision ratios near 98% for both car ads and job ads [ECLS98]. For obituaries, a much more complex challenge, recall ratios ranged from 70% to 100%, and precision ratios ranged from 93% to 100% (except for names of relatives, which dropped to 71%) [ECJ$^+$98]. When challenged to apply our obituary ontology without change to world-wide obituaries from Ireland, Sri Lanka, New Zealand, and India, these results continued to hold, although there was some drop-off caused by cultural localizations that could be corrected within our framework.

Demo Description

Our demo lets a user click on one of the three rectangles in Figure 1 to view an HTML document (see Figures 3 and 4), to view an application ontology (see Figure 5), and to view a populated database (see Figure 6). Clicking on one of the nine internal rectangles in Figure 2 lets a demo user view intermediate results such as cleaned unstructured records or potential attribute-value pairs before heuristic processing.

The pull-down list in Figure 3 (or Figure 4) lets a user select one of 13 stored HTML documents, and the pull-down list in Figure 5 lets a user select one of 11 application ontologies. After choosing an application ontology and an HTML document, a user can click on the "Process the Ontology" button (see Figure 5) to process the chosen application ontology against the chosen HTML document. The results are stored in a relational database against which a user can pose any SQL query (see Figure 6).

An interesting feature of the demo lets users (1) modify a prespecified application ontology, (2) create a new application ontology, or (3) load their own previously created application ontology. The demo also lets users (1) modify HTML pages, (2) load HTML pages from the Web, or (3) load HTML pages stored in their own directory. These capabilities let users apply an ontology to any HTML page of their choosing (even one that is "not appropriate"), modify ontologies to see how they behave, or experiment with their own ontologies.

Our experience in teaching others to use the demo tells us: (1) a new user can begin to do something interesting with a given application ontology in five to ten minutes; (2) we can teach people the syntax and semantics of our ontology-specification language in an hour or two (assuming they understand regular expressions); and (3) users can create interesting ontologies with reasonably good recall and precision ratios in about thirty hours. Indeed, students created several of our demonstration ontologies as a class project in two to three dozen hours of work.

Significance

Our demo addresses the important problem of automatic data extraction. Further, it is particularly significant for database researchers because of its database approach.

Important Problem. People want to know! And so do government agencies, information providers, search-and-retrieval companies, and business-intelligence professionals. But they're swamped with volumes of unstructured data churned out from search engines, corporate Intranets, news feeds, and the increasing global Internet. They want critical information extracted automatically, organized effectively, and presented smartly in personalized information views.

The challenges are huge. Critical information is difficult to locate. Once located, its incompatible formats make it difficult to use effectively. Large volumes of unstructured text must be digested into an easy-to-use, organized, uniform format to support querying, focused searching, and personalized information products.

A Database Approach. Some researchers have argued for a machine-learning approach to data extraction. But our experience with the demo indicates that we can produce handcrafted ontologies that are robust across new and evolving pages with no more human effort than it typically takes to label a training set for machine learning. Further, handcrafted ontologies normally yield higher recall and precision. We are not arguing that machine learning has no place in this technology (it does), but we are arguing, and trying to show with our demo, that database technology can and should play a greater role in resolving these massive information challenges.

Our demo begins to show some possible ways to address these challenges. We apply concepts in conceptual modeling, first-order constraints over database schemes, and knowledge-base ontologies to drive us forward. These long-standing database technologies can provide mechanisms to represent knowledge, store information, and give symbols specific meaning in a particular context. Database technologies can be leveraged to guide, combine, and interpret raw units of information and provide the basis for information extraction, integration, analysis, and presentation.

Cited References of our Work

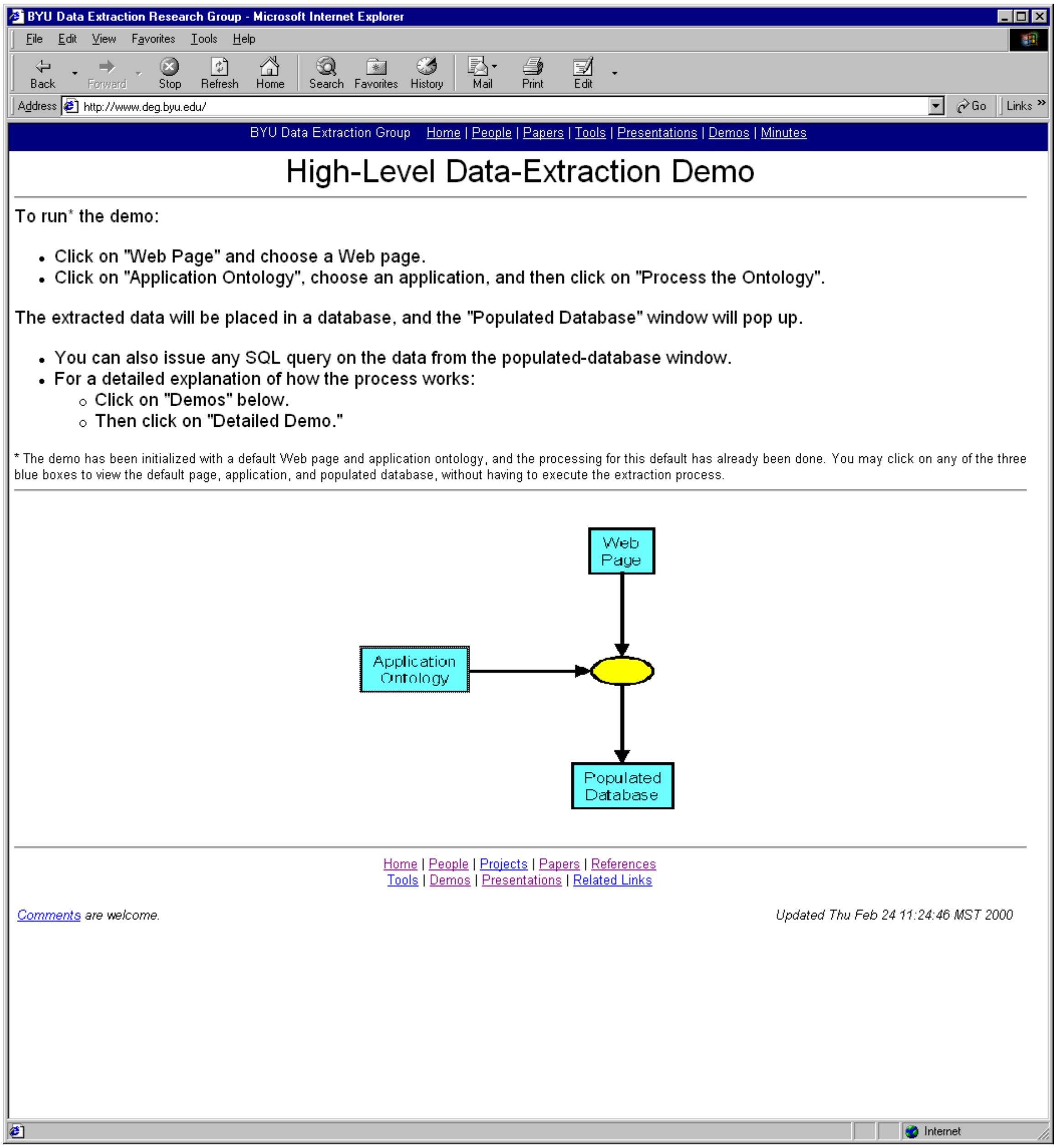
[ECJ+99] D.W. Embley, D.M. Campbell, Y. Jiang, S.W. Liddle, D.W. Lonsdale,Y.-K. Ng, R.D. Smith. Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages. *Data & Knowledge Engineering*, 31(3):227-251, November 1999.

[ECLS98] D.W. Embley, D.M. Campbell, RD. Smith, S.W. Liddle. Ontology-based Extraction and Structuring of Information from Data-Rich Unstructured Documents. In *Proceedings of the 7th International Conference on Information and Knowledge Management* (*CIKM'98*), 52-59, Washington D.C., November 1998.

[ECN+99] D.W. Embley, D.M. Campbell, Y. Jiang, Y.-K. Ng, R.D. Smith, S.W. Liddle, and D.W. Quass. A Conceptual-Modeling Approach to Extracting Data from the Web. In *Proceedings of the 17th International Conference on Conceptual Modeling* (*ER'98*), 78-91, Singapore, November 1998.

[EJN99] D.W. Embley, Y.S. Jiang, and Y.-K. Ng. Record-Boundary Discovery in Web Documents. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (*SIGMOD'99*), 467-478, Philadelphia, Pennsylvania, June 1999.

[Hom00] Home Page for BYU Data Extraction Group, 2000. URL: http://www.deg.byu.edu


BYU Data Extraction Research Group - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://www.deg.byu.edu/
Go Links
BYU Data Extraction Group Home | People | Papers | Tools | Presentations | Demos | Minutes
Web Page
Application Ontology
Populated Database
Internet

High-Level Data-Extraction Demo

To run* the demo:

- Click on "Web Page" and choose a Web page.
- Click on "Application Ontology", choose an application, and then click on "Process the Ontology".

The extracted data will be placed in a database, and the "Populated Database" window will pop up.

- You can also issue any SQL query on the data from the populated-database window.
- For a detailed explanation of how the process works:
  - Click on "Demos" below.
  - Then click on "Detailed Demo."

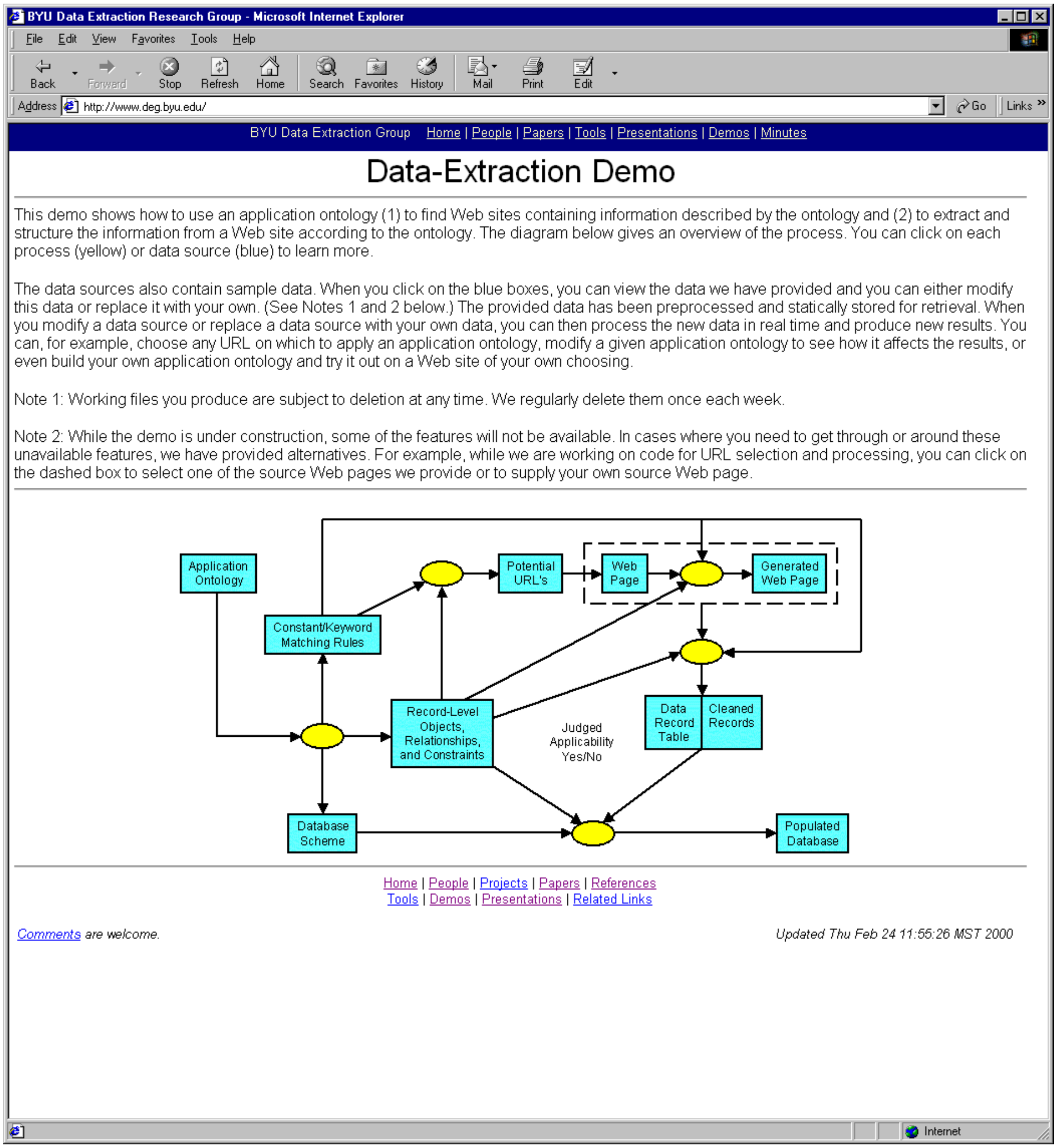
* The demo has been initialized with a default Web page and application ontology, and the processing for this default has already been done. You may click on any of the three blue boxes to view the default page, application, and populated database, without having to execute the extraction process.

Home | People | Projects | Papers | References
Tools | Demos | Presentations | Related Links

Comments are welcome.

Updated Thu Feb 24 11:24:46 MST 2000

Figure 1. High-Level Demo.



BYU Data Extraction Research Group - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Mail
Print
Edit
Address
http://www.deg.byu.edu/
Go
Links
BYU Data Extraction Group
Home | People | Papers | Tools | Presentations | Demos | Minutes

Data-Extraction Demo

This demo shows how to use an application ontology (1) to find Web sites containing information described by the ontology and (2) to extract and structure the information from a Web site according to the ontology. The diagram below gives an overview of the process. You can click on each process (yellow) or data source (blue) to learn more.

The data sources also contain sample data. When you click on the blue boxes, you can view the data we have provided and you can either modify this data or replace it with your own. (See Notes 1 and 2 below.) The provided data has been preprocessed and statically stored for retrieval. When you modify a data source or replace a data source with your own data, you can then process the new data in real time and produce new results. You can, for example, choose any URL on which to apply an application ontology, modify a given application ontology to see how it affects the results, or even build your own application ontology and try it out on a Web site of your own choosing.

Note 1: Working files you produce are subject to deletion at any time. We regularly delete them once each week.

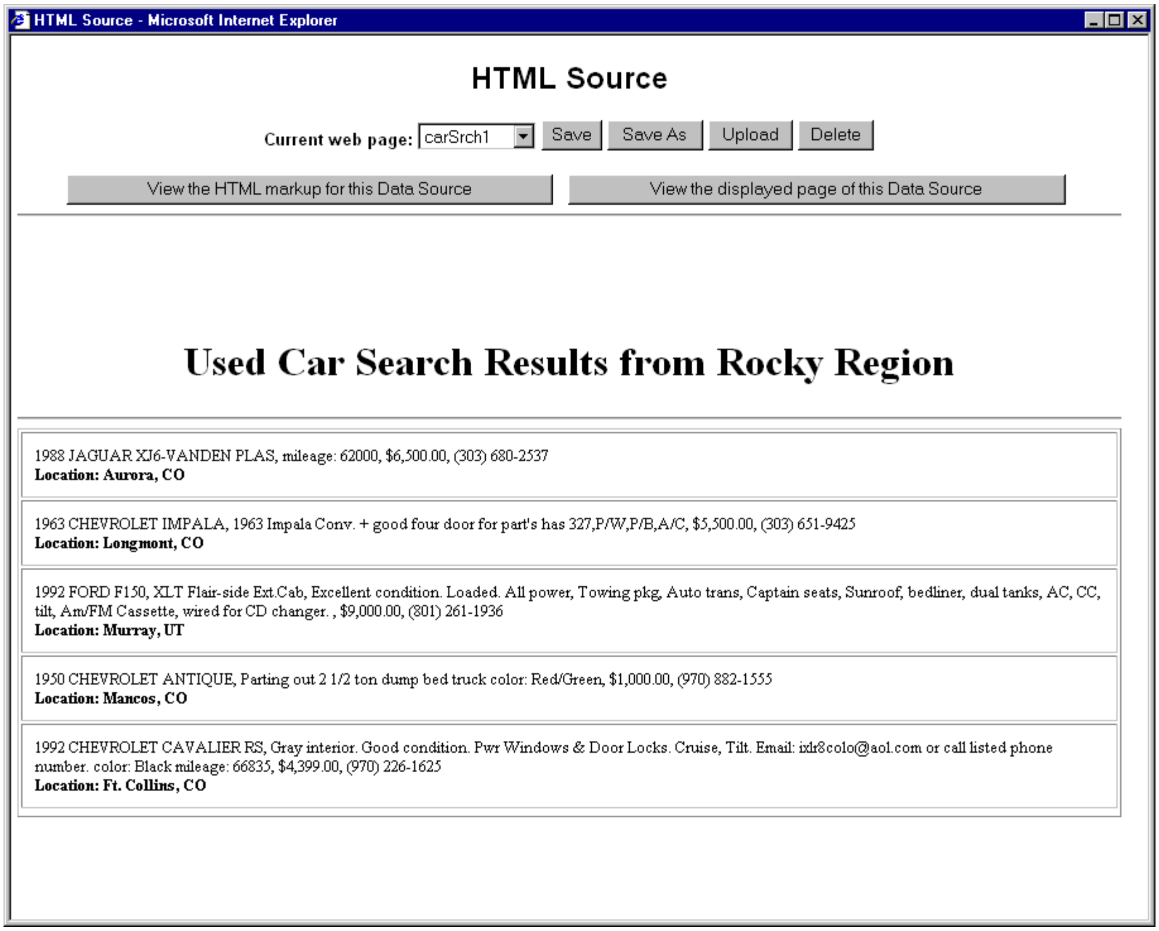
Note 2: While the demo is under construction, some of the features will not be available. In cases where you need to get through or around these unavailable features, we have provided alternatives. For example, while we are working on code for URL selection and processing, you can click on the dashed box to select one of the source Web pages we provide or to supply your own source Web page.

Application Ontology
Constant/Keyword Matching Rules
Potential URL's
Web Page
Generated Web Page
Record-Level Objects, Relationships, and Constraints
Judged Applicability Yes/No
Data Record Table
Cleaned Records
Database Scheme
Populated Database
Home | People | Projects | Papers | References
Tools | Demos | Presentations | Related Links

Comments are welcome.

Updated Thu Feb 24 11:55:26 MST 2000

Figure 2. Detailed Demo.


HTML Source - Microsoft Internet Explorer
Save
Save As
Upload
Delete
View the HTML markup for this Data Source
View the displayed page of this Data Source

HTML Source

Current web page: carSrch1

Used Car Search Results from Rocky Region

1988 JAGUAR XJ6-VANDEN PLAS, mileage: 62000, $6,500.00, (303) 680-2537
Location: Aurora, CO

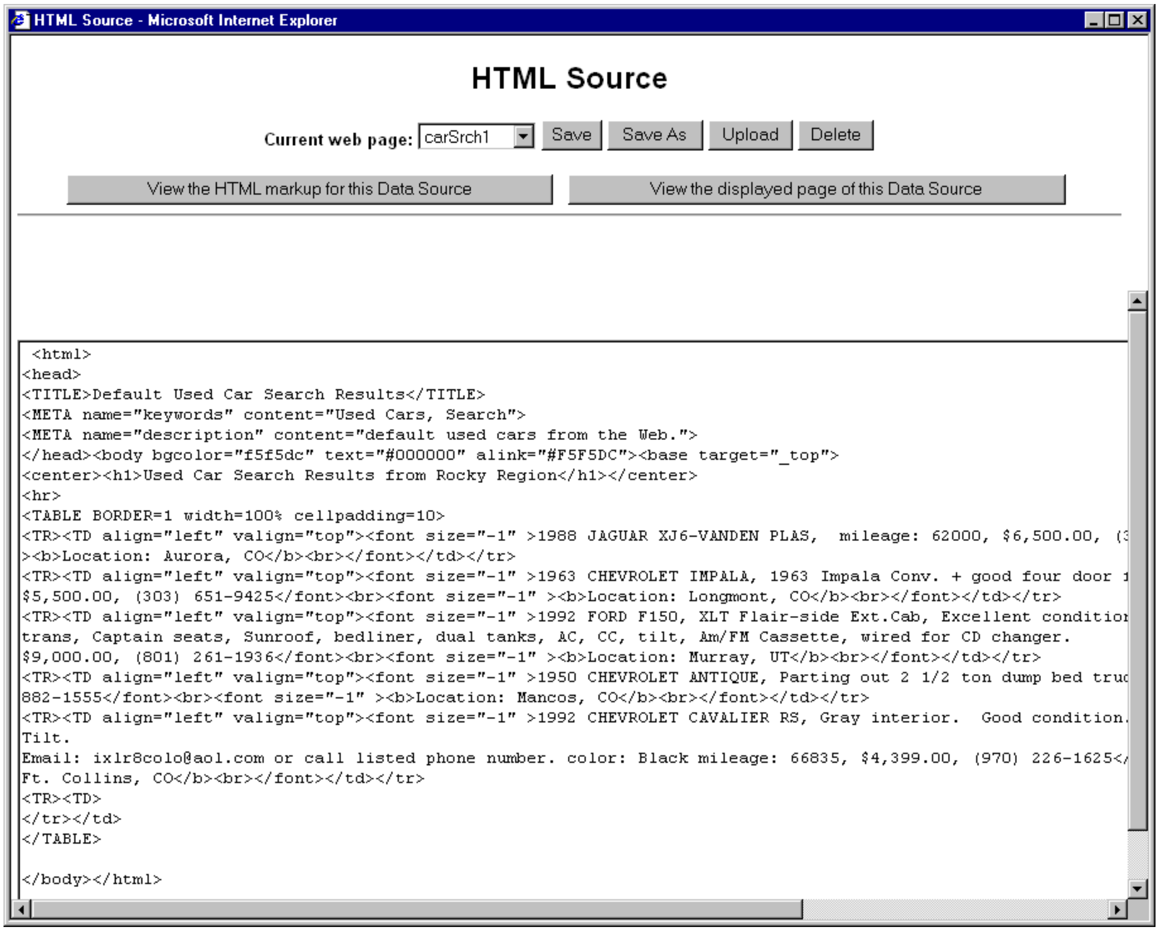
1963 CHEVROLET IMPALA, 1963 Impala Conv. + good four door for part's has 327,P/W,P/B,A/C, $5,500.00, (303) 651-9425
Location: Longmont, CO

1992 FORD F150, XLT Flair-side Ext.Cab, Excellent condition. Loaded. All power, Towing pkg, Auto trans, Captain seats, Sunroof, bedliner, dual tanks, AC, CC, tilt, Am/FM Cassette, wired for CD changer. , $9,000.00, (801) 261-1936
Location: Murray, UT

1950 CHEVROLET ANTIQUE, Parting out 2 1/2 ton dump bed truck color: Red/Green, $1,000.00, (970) 882-1555
Location: Mancos, CO

1992 CHEVROLET CAVALIER RS, Gray interior. Good condition. Pwr Windows & Door Locks. Cruise, Tilt. Email: ixlr8colo@aol.com or call listed phone number. color: Black mileage: 66835, $4,399.00, (970) 226-1625
Location: Ft. Collins, CO

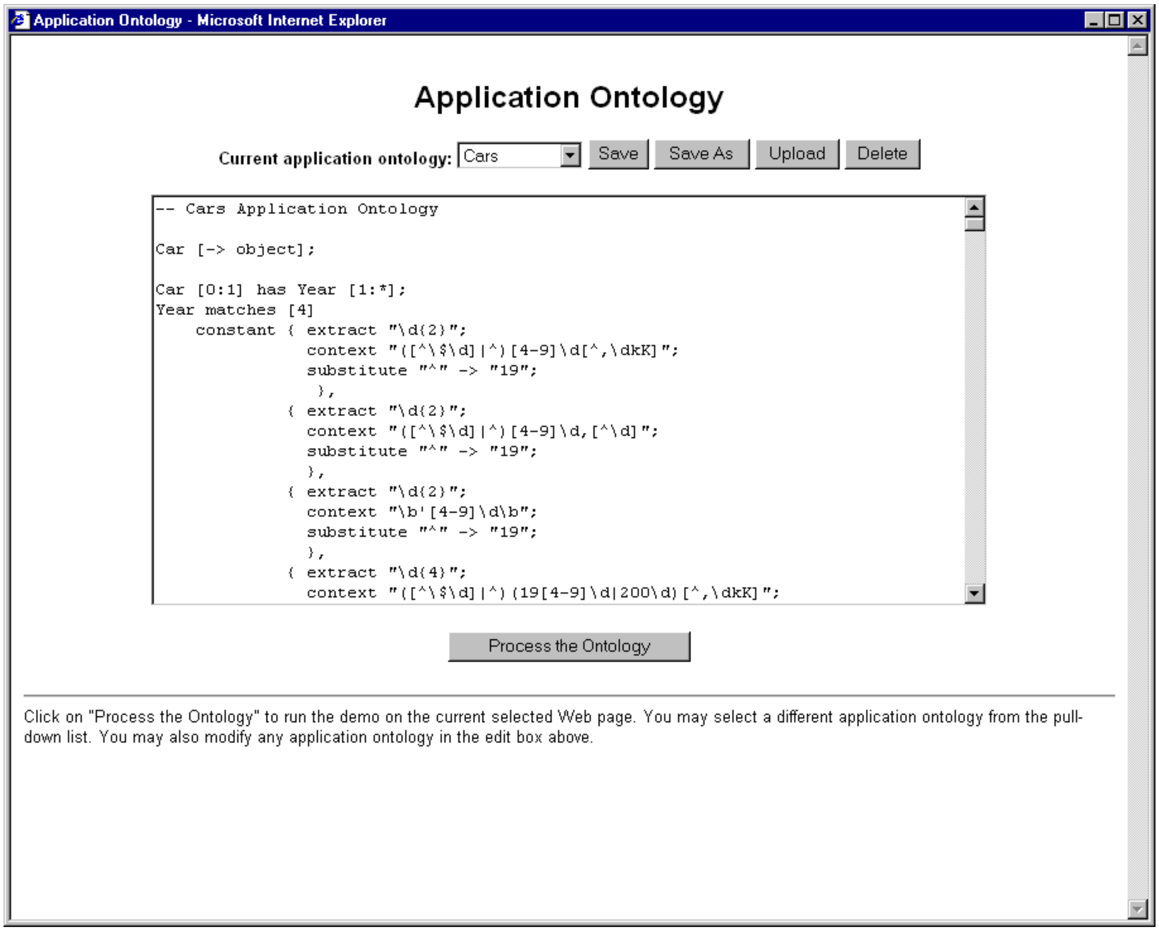
Figure 3. Rendered HTML Source.


HTML Source - Microsoft Internet Explorer
HTML Source
Current web page: carSrch1
Save
Save As
Upload
Delete
View the HTML markup for this Data Source
View the displayed page of this Data Source

```
 <html>
<head>
<TITLE>Default Used Car Search Results</TITLE>
<META name="keywords" content="Used Cars, Search">
<META name="description" content="default used cars from the Web.">
</head><body bgcolor="f5f5dc" text="#000000" alink="#F5F5DC"><base target="_top">
<center><h1>Used Car Search Results from Rocky Region</h1></center>
<hr>
<TABLE BORDER=1 width=100% cellpadding=10>
<TR><TD align="left" valign="top"><font size="-1" >1988 JAGUAR XJ6-VANDEN PLAS,  mileage: 62000, $6,500.00, (3
><b>Location: Aurora, CO</b><br></font></td></tr>
<TR><TD align="left" valign="top"><font size="-1" >1963 CHEVROLET IMPALA, 1963 Impala Conv. + good four door
$5,500.00, (303) 651-9425</font><br><font size="-1" ><b>Location: Longmont, CO</b><br></font></td></tr>
<TR><TD align="left" valign="top"><font size="-1" >1992 FORD F150, XLT Flair-side Ext.Cab, Excellent condition
trans, Captain seats, Sunroof, bedliner, dual tanks, AC, CC, tilt, Am/FM Cassette, wired for CD changer.
$9,000.00, (801) 261-1936</font><br><font size="-1" ><b>Location: Murray, UT</b><br></font></td></tr>
<TR><TD align="left" valign="top"><font size="-1" >1950 CHEVROLET ANTIQUE, Parting out 2 1/2 ton dump bed truc
882-1555</font><br><font size="-1" ><b>Location: Mancos, CO</b><br></font></td></tr>
<TR><TD align="left" valign="top"><font size="-1" >1992 CHEVROLET CAVALIER RS, Gray interior.  Good condition.
Tilt.
Email: ixlr8colo@aol.com or call listed phone number. color: Black mileage: 66835, $4,399.00, (970) 226-1625</
Ft. Collins, CO</b><br></font></td></tr>
<TR><TD>
</tr></td>
</TABLE>

</body></html>
```

Figure 4. HTML Source.



Application Ontology

Current application ontology: Cars

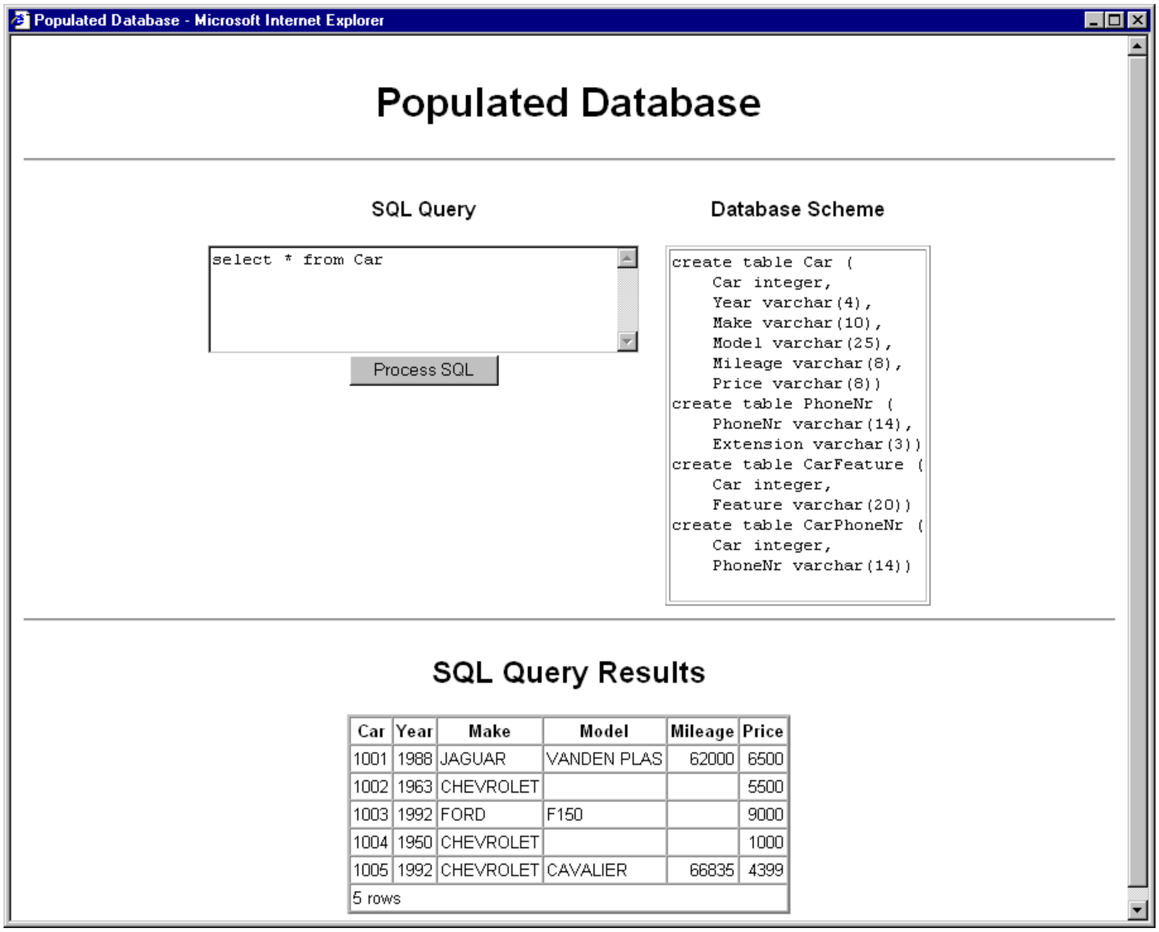
```
-- Cars Application Ontology

Car [-> object];

Car [0:1] has Year [1:*];
Year matches [4]
    constant { extract "\d{2}";
               context "([^\$\d]|^)[4-9]\d[^,\dkK]";
               substitute "^" -> "19";
                },
             { extract "\d{2}";
               context "([^\$\d]|^)[4-9]\d,[^\d]";
               substitute "^" -> "19";
               },
             { extract "\d{2}";
               context "\b'[4-9]\d\b";
               substitute "^" -> "19";
               },
             { extract "\d{4}";
               context "([^\$\d]|^)(19[4-9]\d|200\d)[^,\dkK]";
```

Click on "Process the Ontology" to run the demo on the current selected Web page. You may select a different application ontology from the pull-down list. You may also modify any application ontology in the edit box above.

Figure 5. Application Ontology.



Populated Database

SQL Query

```
select * from Car
```

Database Scheme

```
create table Car (
    Car integer,
    Year varchar(4),
    Make varchar(10),
    Model varchar(25),
    Mileage varchar(8),
    Price varchar(8))
create table PhoneNr (
    PhoneNr varchar(14),
    Extension varchar(3))
create table CarFeature (
    Car integer,
    Feature varchar(20))
create table CarPhoneNr (
    Car integer,
    PhoneNr varchar(14))
```

SQL Query Results

Car	Year	Make	Model	Mileage	Price
1001	1988	JAGUAR	VANDEN PLAS	62000	6500
1002	1963	CHEVROLET			5500
1003	1992	FORD	F150		9000
1004	1950	CHEVROLET			1000
1005	1992	CHEVROLET	CAVALIER	66835	4399
5 rows					

Figure 6. Database Scheme, SQL Query, and Results.