

A Web of Facts: Turning Raw Data into Accessible Knowledge

David W. Embley

Abstract

Based on computational notions of ontology, epistemology, and logic, we offer a way to turn raw data on the web into accessible knowledge—a web of searchable facts, enabling users to directly search for answers to questions. To demonstrate the feasibility of creating a web of facts and enabling ordinary, unsophisticated users to access this knowledge, we plan to build a prototype consisting of (1) a workbench for turning raw data currently found on the web into accessible knowledge, and (2) a query system allowing free-form and form-based queries over this generated knowledge.

1 Introduction

The current web is a web of linked pages. Frustrated users search for facts by guessing which keywords or keyword phrases might lead them to pages in which they can find facts. Can we make it possible for users to search directly for facts? Equivalently, can we turn the web into a web of data (instead of a web of pages containing data)? Ultimately, can the web be a database where users can ask questions and get answers and can query for facts rather than search for pages containing facts?

These questions align with one aspect of the National Science Foundation’s new initiative: Cyber-Enabled Discovery and Innovation (solicitation 07-603). An aspect of this CDI initiative asks: How do we move “from data to knowledge”? How do we “[enhance] human cognition and [generate] new knowledge from [the] wealth of [available] heterogeneous digital data”? In other words (at least from one perspective), how can we turn the web of pages into a web of facts?

To answer these questions, we ask a few more fundamental questions: What is data? What are facts? What is knowledge? How does one know? Philosophers have pursued answers to these questions for centuries, and although we do not pretend to be able to contribute philosophically, we can use their answers to guide us in how to construct a practical knowledge system.

Philosophers study *ontology*, *epistemology*, and *logic*.

- *Ontology* is the study of existence. It answers the question: “What exists?” Computationally, in our quest for turning raw data into accessible knowledge, it answers the questions: “What concepts, relationships, and constraints exist and how are they interrelated?” We answer these questions by providing formal conceptual models of some domain of knowledge, i.e., by declaring the concepts of interest along with the relationships among these concepts and the constraints over these concepts and relationships.

- *Epistemology* is the study of the nature of knowledge. It answers the questions: "What is knowledge?" and "How is knowledge acquired?" Computationally, in our quest, it answers the questions: "Computationally, what is knowledge" and "How does raw data become computationally accessible knowledge?" We answer these questions by populating conceptual models, i.e., by turning raw data into knowledge embedded in the concepts and relationships of interest and according to the declared constraints.
- *Logic* comprises principles and criteria of valid inference. It answers the questions: "What can be inferred?" and "What is known?" Computationally, it answers the questions: "What facts can a query engine infer?" and "What are the known facts (both given and implied)?" We answer these questions by grounding our conceptual model in a description logic—a decidable fragment of first-order logic that describes a collection of facts. To make first-order logic practical for ordinary, unsophisticated users, we must and do add a query generator whose input consists only of ordinary free-form textual expressions or ordinary fill-in-the-blank query forms.

To actualize these ideas, we propose a way to turn raw symbols contained in web pages (or other source documents) into knowledge and to make this knowledge accessible by ordinary people via the web.

2 From a Web of Pages to a Web of Facts

We use an example to show how we propose to turn a web page into a page of queryable data. Figure 1 shows part of two ordinary, human-readable web pages about cars for sale. The facts in these pages are straightforward: A '93 NISSAN is for sale; it is sweet cherry red, has air conditioning, and sells for \$900. Facts on other pages are much less straightforward for many people (e.g., Figure2), but a specialist can see a myriad of facts: Chromosome 17 starts at location 1,194,558, ends at 1,250,267, and has 55,709 bases. Users would like to be able to query the facts on these pages directly: "Find me a red Nissan for under \$5000; it should be a 1990 or newer and have less than 120K miles on it." Or, "Find the location and size of chromosome 17." We cannot, however, directly access these facts. Our proposal makes these facts visible from outside the page and directly accessible to query engines (as opposed to search engines).

2.1 From Symbols to Knowledge—Ontological and Epistemological Tools

To make facts available to query engines, we first map out a guiding pathway to turn raw symbols into knowledge. *Symbols* are characters and character-string instances (e.g. \$, red, chromosome, 55,709). *Data* builds on *symbols* by adding conceptual meta-tags (e.g. Price: \$900, Color: red,

City Weekly SALT LAKE WWW.SLWEEKLY.COM
 news | arts & entertainment | event listings | dining listings | classifieds | best of utah | about...

City Weekly Classifieds

enter keywords here search [Place your ad in the City Weekly classifieds! 801-575-7028 | email](#)

[Backstop](#) | [Adult](#) | [I Saw You](#) | [Help Wanted](#) | [Autos](#) | [Services](#) | [Buy, Sell, Trade](#) | [Legal Notices](#) | [Music](#) | [Mind, Body, Spirit](#) | [Rentals](#) | [Roommates](#) | [Business](#) | [Real Estate](#) | [Real Estate Services](#)

Classifieds: Autos
 16–24 from 24 results.
[< previous 15](#)

'97 MITSUBISHI
 Montero LS, white with beige trim, great condition, 160K miles, 4X4, CD, AC, trailer hitch, \$4700 OBO. 801-244-6404

'93 NISSAN
 Model XE, \$900, Air Conditioning, new tires, sweet cherry red. For listings call 1-800-749-8104 ext. V896. Fee(ucan)

'97 SAAB
 900 SE Talledega 103k mi. 4 n
 Runs GREAT! \$7000 OBO. 801-

'99 DODGE
 RAM 1500 QUAD CAB. 4X4, SL
 offroad tires, lift kit, great cond

'99 PORSCHE
 911, Carrera, mint condition, 5
 sacrifice \$39,900. Call 209-213

'99 VOLKSWAGON
 Black Jette GL, 2 door, minor d

CLASSIC 1966
 CHEVY BELAIR, 4 door, 283 V8
 up, NEW belts, brakes, hoses,

March 14
 397 vehicles
 132 jobs
 176 homes

OnlineAthens
 ATHENS BANNER-HERALD

BONA FIDE CLASSIFIED

Athens, GA **NEWS | SPORTS | DOGBYTES | ROCKATHENS | CLASSIFIEDS | JOBS | HOMES | AUTOS | CITY GUIDE**

TRANSPORTATION

(385 results) - Displaying 1-25

Previous Page [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) Next Page

Price	Year	Make & Model	Description
\$2,000	1984	Dodge	DODGE W100 1984. 4x4 Pickup. 6" lift 12.5x35" mud tires. Runs good. Good hunting truck. \$2,000 cash. 706-769-4466. Add to My List
\$19,800	2002	TOYOTA TUNDRA	TOYOTA 4WD V8 2002, SR5 Tundra, regular cab. 8' bed. Loaded with upgrades. 100k warranty. Line-X. \$19,800. 706-769-4323. Add to My List
\$2,550	1982	CHEVROLET BLAZER	CHEVROLET BLAZER SILVERADO K5 1982. 4x4. 4 speed. Full size. Black. Cold AC. 350 V8. Tow package w/brakes. Tape. Looks & runs great. Only 155K mi. \$2,550. 706-372-6579 or 706-540-0939. Add to My List
\$3,450	1986	FORD BRONCO	FORD BRONCO 1986, 302 engine, 4 wheel drive, 116k miles, good condition, runs good. \$3,450 negotiable. Call 706-367-9061. Add to My List
\$4,500	1993	NISSAN	NISSAN SE-V6, 1993, 4x4, ext cab, 5 spd, camper shell, bed liner, CD, cruise, AC, good tires, only 117K, great shape, but runs rough, must sell, \$4,500 obo, 706-207-8033. Add to My List

Classifieds
 Real Estate Sales
 Real Estate For Rent
 Employment
 Financial
 Transportation
 Rec. Vehicles
 Merchandise
 Garage/Yard Sales
 Agricultural
 Pets & Livestock
 Personals
 Announcements
 Legal Notices
 Service Directory
 Marketplace
 Homes
 Jobs
 Autos
 Business Directory
 OnlineAthens
 News
 UGA News
 Obituaries
 Police Central
 Sports

Figure 1: Sample Car Ads Web Pages.

Size: 55,709). *Conceptualized data* groups data tagged with conceptual identifiers into the framework of a conceptual model (e.g., the conceptual model of cars for sale in Figure 3, the conceptual model of a tiny part of the world of proteins in Figure 4). We have *knowledge* when we populate a conceptual model with correct¹ conceptualized data.

To specify ontological descriptions, we need a conceptual-modeling language. We use OSM [EKW92], which lets us classify and describe things that exist as object sets, relationships among these things as relationship sets, and constraints over these object and relationship sets. As it turns out, OSM is equivalent to an *ALCN* description logic [BN03] in its formalism, which gives

¹*Correct* is interesting. How do we know whether conceptualized data is correct? Humans struggle to know; machines will never know. For the system we are proposing, we rely on evidence and provenance by always linking conceptualized data back to its original source, the human-readable web page from which it was extracted.

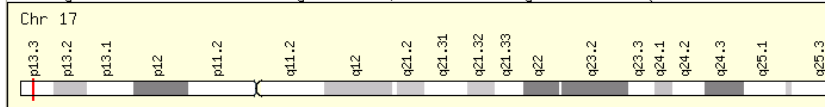

<p>Genomic Location (According to GeneLoc and/or HGNC, and/or Entrez Gene (NCBI build 36), and/or miRBase, Genomic Views According to UCSC and Ensembl) About This Section</p> <p>Jump to Section... ▾</p>	<p>Chromosome: 17 Entrez Gene cytogenetic band: 17p13.3 Ensembl cytogenetic band: 17p13.3</p> <p>Gene in genomic location: bands according to Ensembl, locations according to GeneLoc (and/or Entrez Gene and/or miRBase)</p>  <p>GeneLoc gene densities for chromosome 17</p> <p>GeneLoc location for GC17M001194: (about GC identifiers) Start: 1,194,558 bp from pter End: 1,250,267 bp from pter Size: 55,709 bases Orientation: minus strand</p> <p>Exon Structure </p> <p>1 alternative location: Chr7+,CRA_TcAGchr7v2 63,208,480-63,232,567</p> <p>RefSeq genomic assemblies: NT_079593.2 NC_000017.9 NT_010718.15</p> <p>Genomic View: UCSC Golden Path with GeneCards custom track</p>
<p>Proteins (According to UniProt, and/or Ensembl, Phosphorylation sites)</p>	<p>UniProt/Swiss-Prot: 1433E_HUMAN_P62258 (See protein sequence) Size: 255 amino acids; 29174 Da Subunit: Homodimer. Interacts with NDEL1 (By similarity). Interacts with HCV core protein Subcellular location: Cytoplasm (By similarity). Melanosome. Note=Identified by mass spectrometry</p>

Figure 2: Sample Molecular-Biology Web Page.

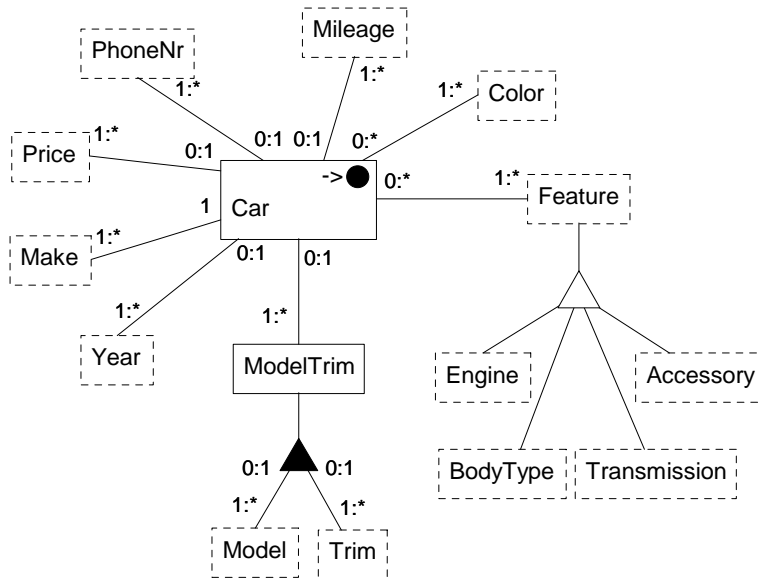


Figure 3: Ontology of Cars for Sale.

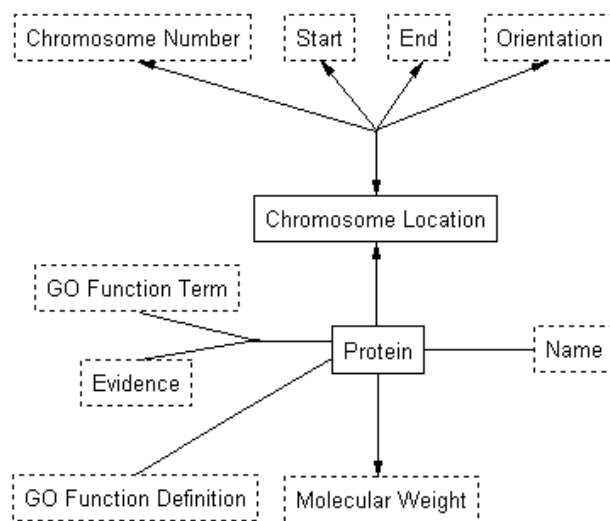


Figure 4: Ontology of a Tiny Part of the World of Proteins.

it the formal properties it needs both for storing and querying knowledge.

A necessary basic tool is an editor that allows users to construct conceptual models such as the ones in Figures 3 and 4. Building ontologies by hand, however, is tedious and is a bottleneck in the process of turning data into knowledge. Can we automatically construct an ontology for a domain of knowledge from raw source domain information? If so, how?

Attempts to extract ontologies from natural-language text documents have been unsuccessful. Although the jury is still out, attempts to extract ontologies from semi-structured documents, such as the ones in Figures 1 and 2, appear promising. Thus, we build tools to use the very web pages we wish to turn into knowledge as sources to help us construct an ontology. This works by recognizing that the data is formatted in a particular way (e.g., the table in the OnlineAthens ads in Figure 1) and by using reverse-engineering techniques to construct a conceptual model (e.g., to discover that *Price*, *Year*, *Make*, and *Model* in the table in Figure 1 are object-set concepts for the conceptual model in Figure 3).

Since we cannot fully count on these automatic ontology-building tools, we also provide a way to build ontologies that leverages the idea of an ordinary form. People generally know how to design forms such as the one in Figure 5. We can algorithmically turn forms into conceptual models. When processed algorithmically, the form in Figure 5 becomes the conceptual model in Figure 4.

Although we can cross the barrier of building conceptual models, these ontological descriptions are not enough. We also need a way to link raw facts in web pages with ontological descriptions. We need epistemological tools as well as ontological tools.

A way to link the actual facts with an ontology is to annotate a web page with respect to

The form is titled "Protein" and contains the following sections:

- Name:** A single-line text input field.
- Molecular Weight:** A single-line text input field.
- GO Function Term and Evidence:** Two side-by-side text input fields.
- GO Function Definition:** A single-line text input field.
- Chromosome Location:** A single-line text input field.
- Chromosome Data Table:** A table with four columns: "Chromosome Number", "Start", "End", and "Orientation". Each column has a corresponding text input field below it.

Figure 5: Sample Form.

an ontology. A data value V in a web page annotated for an object set S in an ontology O is a mapping from V to S in O . Likewise, we annotate related pairs (and in general related groups) of values in a web page by mapping them to a relationship set in an ontology. Figure 6 shows a screen-shot of an annotation tool we have built. The top part of the figure is part of the OnlineAthens web page in Figure 1. Within the page the value $117K$ is highlighted. In the demo we have built, when a user causes the mouse cursor to hover over an annotated value (over the annotated value $117K$ in Figure 6), the demo system highlights the value and displays the connecting link to the ontology. Although unreadable in Figure 6, the link is to the *Mileage* object set in the ontology in Figure 3. The bottom part of Figure 6 shows all the values that have been annotated, with single values for a car such as *Make* displayed and multiple values such as the *Accessory* list hidden under a *Show* button. On the far right, the *Source* column links back into the original page; clicking on this link allows a user to see the car ad in the original source page from which the values for the car have been extracted.

Although it is possible to annotate a web page by hand with respect to an ontology, this is too tedious and time consuming to be practical. We have therefore augmented ontologies with instance recognizers (ontologies augmented with instance recognizers are called *extraction*

The screenshot shows a web page with a sidebar menu on the left and a main content area. The sidebar menu includes: Personals, Announcements, Legal Notices, Service Directory, Marketplace, Homes, Jobs, Autos, Business Directory, OnlineAthens, News, UGA News, Obituaries, Police Central, Sports, DogBytes, Prep Sports, Features, RockAthens, Entertainment, Opinion, Forums, and Contact Us. The main content area displays a list of car advertisements with columns for Price, Year, Make, Model, and Description. Each advertisement has an 'Add to My List' button. A yellow box highlights a URL: `C:\testfiles\carads_ann_ov\#Mileage_<CarAds->Mileage`. Below the list is a summary table with columns: Make, Model, Trim, Year, Mileage, PhoneNr, Price, Color, Transmission, Engine, BodyType, Accessory, OtherFeature, and Source. The summary table contains seven rows of data corresponding to the advertisements above.

Make	Model	Trim	Year	Mileage	PhoneNr	Price	Color	Transmission	Engine	BodyType	Accessory	OtherFeature	Source
Dodge			1984		706-769-4466	2,000				Show			5
TOYOTA			2002	100k	706-769-4323	19,800			Show	Show			7
CHEVROLET			1982	155K	706-372-6579	2,550	Show		Show	Show			9
FORD			1986	116k	706-367-9061	3,450							11
NISSAN		SE	1993	117K	706-207-8033	4,500		Show	Show	Show	Show		13
Ford	EXPLORER		1998	100K	706-769-3060	5,100	Show	Show	Show	Show	Show		15
CHEVROLET			1971	18K	706-357-5145	16,000	Show						17

Figure 6: Annotated Web Page.

ontologies). Instance recognizers contain regular expressions that recognize common textual items such as dates, times, prices, and numbers. They also contain lexicons that match with items such as car makes and models and protein names and functions. Much can and has been said about how to build and use these instance recognizers embedded within extraction ontologies (e.g., [ECJ⁺99, DEL06] among several others).

Building instance recognizers is itself tedious and time consuming. We have four answers to this legitimate observation: (1) We have a library of common instance recognizers that can be used directly or specialized for use in some domain. (2) We need not have perfect recognizers because we can augment extraction ontologies with pattern-based extractors—once a few values have been recognized, pattern-based extractors can determine the pattern of a semi-structured page and thereby correctly recognize values beyond those observed by instance recognizers. (3) We can automatically update lexicon recognizers by adding additional values found by pattern-based extractors. And (4) neither instance recognizers nor lexicons are absolutely necessary because we

```

<rdf:RDF ... xmlns:ontos="http://www.deg.byu.edu/ontology/ontosBasic#"
          xmlns:carad="http://www.deg.byu.edu/ontology/carad#"
          xmlns:webpage="http://www.deg.byu.edu/demos/..." ... >
...
  <rdf:Description rdf:about="&webpage;CarIns13">
    <carad:Mileage>117000</carad:Mileage>
    <carad:Price>4500</carad:Price>
    <carad:Make>Nissan</carad:Make>
    <carad:Year>1993</carad:Year>
  ...
</rdf:Description>
  <rdf:Description rdf:about="&webpage;Mileage13">
    <ontos:ValueInText>117K</ontos:ValueInText>
    <ontos:CanonicalValue>117000</ontos:CanonicalValue>
    <ontos:CanonicalDatatype>xsd:integer</ontos:CanonicalDatatype>
    <ontos:CanonicalDisplayValue>117,000</ontos:CanonicalDisplayValue>
    <ontos:Offset> 37733 </ontos:Offset>
  </rdf:Description>
...
</rdf:RDF>

```

Figure 7: RDF Triples Annotating a Car Ads Web Page.

can let users provide a few sample mappings from a page to an ontology, and from these sample mappings generate pattern-based extractors.

2.2 Querying Knowledge—Logic Tools

After building tools to turn raw symbols in web pages into knowledge, we next need to provide appropriate query capabilities. Given that we have data on a web page annotated with respect to an ontology, we can immediately generate RDF² data like the tagged data in Figure 7. We can then directly use the SPARQL query language [W3Cb] to write and execute queries over this RDF data. Figure 8 shows a SPARQL query over the RDF data that searches for red Nissans whose price is less than \$5,000, whose year is after 1990, and whose mileage is less than 120K.

If ordinary people in everyday situations could write SPARQL queries, we would basically have what we need to enable users to search the web of facts. Unfortunately, ordinary users are unwilling to learn SPARQL (indeed, most are likely incapable of learning SPARQL).

We therefore need to provide a query system in which users can pose queries using their own terms. Figure 9 shows a screen-shot³ of a free-form query tool we have built [AME06, AME07]. The key to making these free-form queries work is not natural-language processing (at least not in the usual sense of natural-language processing), but is rather to apply extraction ontologies

²RDF stands for *Resource Description Framework*. RDF is a W3C standard for representing ontological data [W3Ca].

³Note that the query in the screen-shot is the query about red Nissan’s posed as an example in the introduction.


```

PREFIX carad: <http://www.deg.byu.edu/ontology/carad#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?make ?color ?price ?year ?mileage
WHERE { ?x carad:Make ?make . FILTER (?make = "Nissan") .
  OPTIONAL {?x carad:Color ?color} . FILTER (?color = "red") .
  OPTIONAL {?x carad:Price ?price} . FILTER (xsd:integer(?price) < 5000) .
  OPTIONAL {?x carad:Year ?year} . FILTER (xsd:integer(?year) >= 1990) .
  OPTIONAL {?x carad:Mileage ?mileage} .
  FILTER (xsd:integer(?mileage) < 120000) }

```

Figure 8: SPARQL Query to Search for a Red Nissan.

to the queries themselves. This lets us align user queries with ontologies and thus with facts in annotated web pages.

Anyone can readily pose free-form queries. To be successful, however, a user does, however, have to guess what keywords, values, and constraint expressions might be available in an extraction ontology for the domain of interest. This is similar to users having to guess keywords and values for current search-engine queries. Since users may not always be successful at asking free-form queries, we also provide a form query language, based on a domain ontology, that allows a user to fill it out and submit it to pose queries in much the same way as users currently pose queries by filling in forms on the current web (e.g., the form in Figure 10). Interestingly, these query forms are automatically derivable from domain ontologies, and thus need not be specified by developers. Instead of reverse-engineering a form like the one in Figure 5 to the ontology in Figure 4, we can forward-engineer (derive) the form from the ontology and use it in a natural-forms query language [Emb89].

Finally, we we must make all of this scale globally. Although we have thought about this to some degree (see [AMELT07]), we have only begun to wrestle with the practical difficulties. We have defined semantic indexing, with which we can quickly find applicable ontologies for user queries, and we have considered large-scale caching, following Google’s lead. We realize, however, that we will not be able to achieve this on our own. But we can show the way and provide the technical answers to make it all work.

3 Research Plan

The work proposed here presents a grand vision—not something that can be accomplished in a few short months. We have already accomplished much, but much remains to be done. The grand vision, however, is within sight.

Basically, the next steps are to finish building and testing some of the major components and to assemble all the tools into a workbench prototype. Building and testing are well underway, and

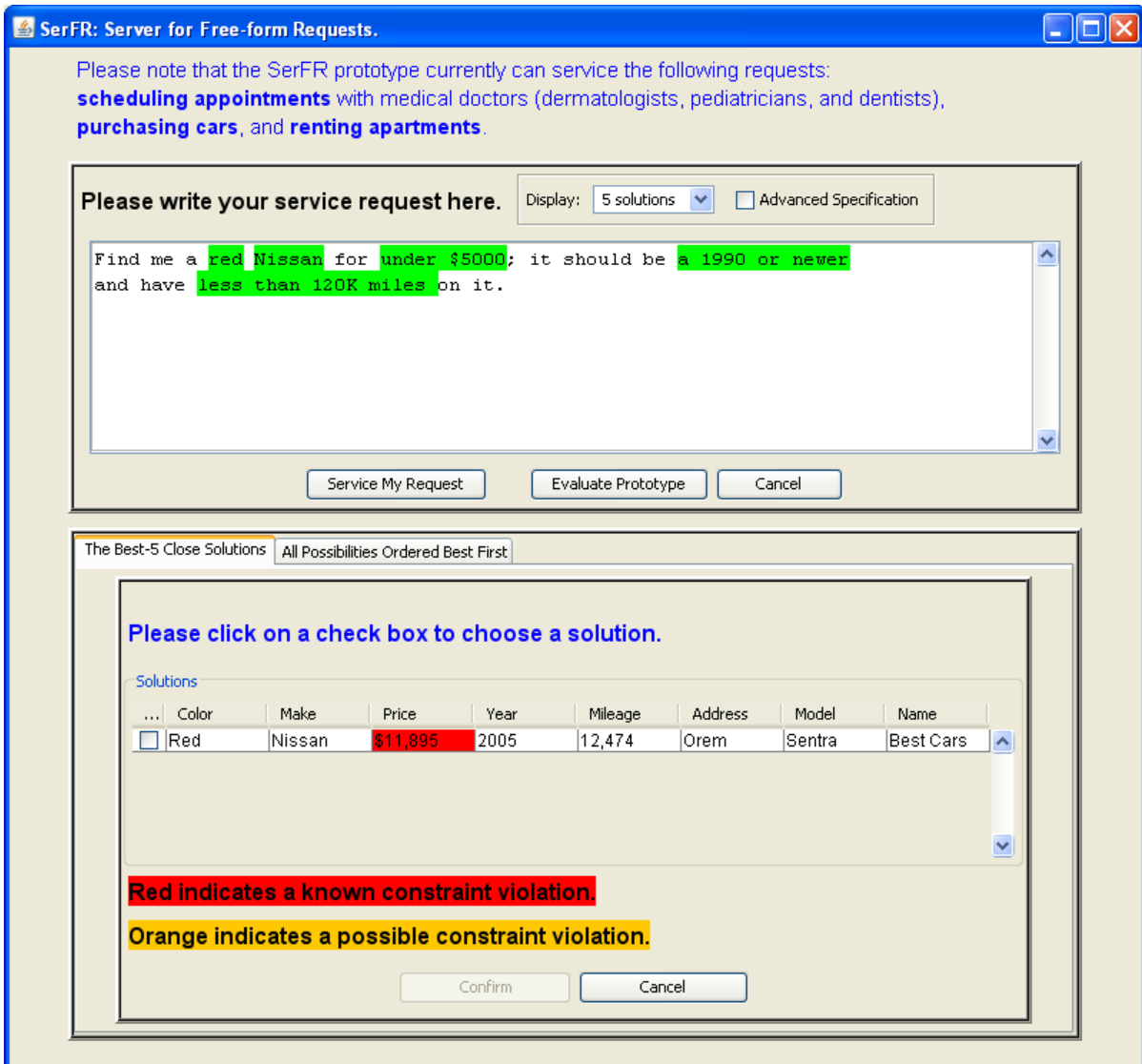


Figure 9: Free-form Query for Red Nissans.

it appears that we will achieve hypothesized results. Tool assembly, however, is nontrivial, and making everything work together, as envisioned, is likely to be tough. Although tough, the work does not require an exploration of yet unknown paths. Rather, it requires understanding the tools as implemented, the vision as explained, and a strong ability to program, specifically to program in Java, and to work with integrated development environments, specifically Eclipse. Our upper-level undergraduate computer science students are capable of doing this work and developing a prototype showing that the grand vision is technically feasible.

As the budget section of the MEG grant shows, two undergraduate students are to join the already existing mentoring environment and take on as their task the development of the proof-of-concept prototype by integrating existing project code into one complete whole. These undergrad-

Search Vehicles
Inventory last updated:
Tuesday, November 20, 2007

Search for: * Required

Make:

Vehicle Type:

Model:

Price:

SEARCH >

Figure 10: Typical Web Form for a User Query.

uates will be able to rub shoulders with experienced graduate students—both MS and PhD—and will have the opportunity to see an exciting research prototype unfold.

4 Expected Significance

Intellectual Merit. The research work:

- provides an answer to the question about how to turn syntactic symbols into semantic knowledge;
- shows how to create a web of facts;
- explores the synergistic interplay among ontology, epistemology, and logic for the advancement of knowledge, providing new ways to think computationally about what knowledge is and how knowledge is acquired; and
- provides a way for ordinary, unsophisticated users to query and reason over fact-filled ontologies.

Broader Impact. The research work has the potential to help people:

- harvest and make available facts from the wealth of available heterogeneous digital data;
- harness and manage community knowledge with the objective of enhancing human cognition;
- make facts on the web (rather than pages) easily searchable by the general public;
- provide a practical set of tools for knowledge management; and
- involve knowledge workers from various disciplines in a community-wide effort to convert data into knowledge.

References

- [AME06] M.J. Al-Muhammed and D.W. Embley. Resolving underconstrained and overconstrained systems of conjunctive constraints for service requests. In *Proceedings of the 18th International Conference on Advanced Information Systems Engineering (CAiSE'06, LNCS 4001)*, pages 223–238, Luxembourg City, Luxembourg, June 2006.
- [AME07] M. Al-Mumammed and D.W. Embley. Ontology-based constraint recognition for free-form service requests. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE 2007)*, pages 366–375, Istanbul, Turkey, April 2007.
- [AMELT07] M.J. Al-Muhammed, D.W. Embley, S.W. Liddle, and Y. Tijerino. Bringing web principles to services: Ontology-based web services. In *Proceedings of the Fourth International Workshop on Semantic Web for Services and Processes (SWSP'07)*, pages 73–80, Salt Lake City, Utah, July 2007.
- [BN03] F. Baader and W. Nutt. Basic description logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The Description Logic Handbook*, chapter 2, pages 43–95. Cambridge University Press, Cambridge, UK, 2003.
- [DEL06] Y. Ding, D.W. Embley, and S.W. Liddle. Automatic creation and simplified querying of semantic web content: An approach based on information-extraction ontologies. In *Proceedings of the First Asian Semantic Web Conference (ASWC 2006)*, pages 400–414, Beijing, China, September 2006.
- [ECJ+99] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering*, 31(3):227–251, November 1999.
- [EKW92] D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object-oriented Systems Analysis: A Model-Driven Approach*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.
- [Emb89] D.W. Embley. NFQL: The natural forms query language. *ACM Transactions on Database Systems*, 14(2):168–211, June 1989.
- [W3Ca] Resource Description Framework (RDF). www.w3.org/RDF. W3C (World Wide Web Consortium).
- [W3Cb] SPARQL Query Language for RDF. www.w3.org/TR/rdf-sparql-query. W3C (World Wide Web Consortium).