Chapter 15
Conceptual Modeling Foundations for a Web of Knowledge

David W. Embley and Stephen W. Liddle and Deryle W. Lonsdale

Abstract The semantic web purports to be a web of knowledge that can answer our questions, help us reason about everyday problems as well as scientific endeavors, and service many of our wants and needs. Researchers and others expound various views about exactly what this means. Here we propose an answer with conceptual modeling as its foundation. We define a web of knowledge as a collection of interconnected knowledge bundles superimposed over a web of documents. Knowledge bundles are conceptual model instances augmented with facilities that provide for both extensional and intensional facts, for linking between knowledge bundles yielding a web of data, and for linking to an underlying document collection providing a means of authentication. We formally define both the component parts of these augmented conceptual models and their synergistic interconnections. As for practicalities, we discuss problems regarding the potentially high cost of constructing a web of knowledge and explain how they may be mitigated. We also discuss usage issues and show how untrained users can interact with and gain benefit from a web of knowledge.

15.1 Introduction

Ideas about the semantic web have been with us ever since Tim Berners-Lee's published his book, *Weaving the Web* [BL99], and his *Scientific American* article, *The Semantic Web* [BLHL01], with Hendler and Lassila. They and others have continued to discuss these ideas in an effort to more fully explain the semantic-web vision—its practicalities, successes, and challenges

David W. Embley, Stephen W. Liddle, Deryle W. Lonsdale
Brigham Young University, Provo, Utah 84602, USA, e-mail: embley@cs.byu.edu, liddle@byu.edu, lonz@byu.edu

[SHBL06, AH08]. The W3C web site introduces the semantic web simply as "a web of data" [W3C].

Many of these ideas hark back even to the days of Plato [PlaBC] and Aristotle [AriBC] and the beginnings of philosophical discussions about ontology, epistemology, and logic. Here, we begin with this ancient view of semantics and show how it leads to a view of the semantic web rooted in conceptual modeling. In particular, we show how conceptual modeling can unify a view of these fundamental concepts and provide a practical way to realize them. Our intent is not to resolve questions about what the semantic web is, but rather to provide a practical view of one possible path toward realizing some of the benefits claimed by semantic-web visionaries. We call our conceptual-modeling view of the semantic web a "Web of Knowledge" (a "WoK").

To motivate our vision of a WoK, consider the current web of pages, which contains a wealth of knowledge. Unfortunately, most of the knowledge is not encoded in a way that enables direct user query. We cannot, for example, directly google for a car that is a 2003 or newer selling for under 15 grand; or for the names of the parents of great-grandpa Schnitker; or for countries whose population will likely decrease by more than 10% in 50 years. A way to enable direct query for facts embedded in web pages and facts implied by these stated facts is to annotate facts with respect to ontologies. Annotating facts implicitly populates these ontologies, turning them into a database over which structured queries can be executed. Annotation links also provide a form of provenance and authentication, allowing users to verify query results by checking original sources. Furthermore, facts and ontological concepts may appear in more than one populated ontology. Linking facts and ontological concepts across ontologies can provide navigation paths to explore additional, related knowledge. The web with a superimposed layer of interlinked ontologies each annotating a myriad of facts from the underlying web becomes a *Web of Knowledge*, a *WoK*.

Although this vision of a WoK is appealing, there are significant barriers preventing both its creation and its use. Ontology languages exist, with OWL being the de facto standard. RDF files can provide data for these ontologies and can also store annotation information linking data to facts in web pages and linking equivalent information in RDF files to one another. The SPARQL query language is a standard for querying RDF data. SWRL rules can provide for reasoning. Thus, all constituent components for a WoK are W3C standards in common use, and they even all work together allowing for immediate WoK development and usage. Nevertheless, the barriers of creation and usage remain high and effectively prevent WoK deployment. The creation barrier is high because of the cost involved in developing OWL ontologies and annotating web pages by linking RDF-encoded facts in web pages to these OWL ontologies. The usage barrier is high because untrained users cannot write SPARQL queries and SWRL rules.

In this exposition, we show how conceptual modeling can enable a WoK—can provide a firm foundation for a WoK and ways to break through the barriers to WoK creation and usage. We begin in Section 15.2 by discussing a computational view of ontology, epistemology, and logic. We argue that conceptual models, augmented in a particular way, build nicely upon these philosophical ideas so as to enable a WoK. In Section 15.3 we formalize this foundation. The formalization leads to a clear understanding of what must be done to create a WoK and make it usable. We then discuss initiatives we have investigated to address these challenges and opportunities—for construction in Section 15.4 and for usage in Section 15.5. We conclude in Section 15.6.

15.2 WoK Conceptualization

To think about constructing and using a web of knowledge, we first ask some fundamental questions: What is data? What are facts? What is knowledge? How does one reason and know? Philosophers have pursued answers to these questions for millennia; and although we do not pretend to be able to contribute to philosophy, we can use their ideas about ontology, epistemology, and logic to guide us in how to build and use a WoK.

- *Ontology* is the study of existence. It asks: "What exists?" In our quest to build a WoK, we must find computational solutions the question: "What concepts, relationships, and constraints exist?" We answer computationally, saying that we can declare a formal conceptual model for some domain of knowledge that captures the relevant concepts along with the relationships among these concepts and the constraints over these concepts and relationships.[1]
- *Epistemology* is the study of the nature of knowledge. It asks: "What is knowledge?" and "How is knowledge acquired?" To build a WoK, we provide computational answers to "What is digitally stored knowledge?" and "How does raw data become algorithmically accessible knowledge?" Our answer is to turn raw data into knowledge by populating conceptual models—by embedding facts in the concepts and relationships in accord with constraints. We further follow Plato's lead in wanting our knowledge to be justified [PlaBC], and thus we provide (1) annotation links that connect facts embedded in ontologies to sources from which they are extracted and (2) data and concept "same-as" connections that link objects and concepts across populated ontologies.
- *Logic* comprises principles and criteria of valid inference. It asks: "What is known?" and "What can be inferred?" In the computational context

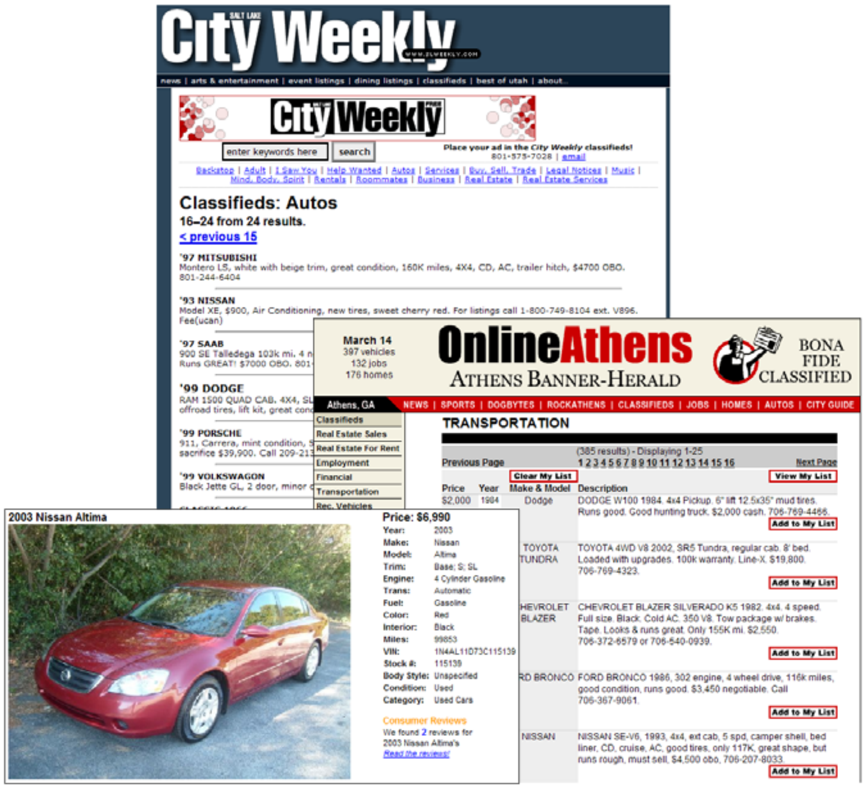
[1] Purists argue that conceptual models are not ontologies [Gru93, Gua98, Smi03]. We agree that when conceptual models play their traditional role to aid in database schema design, they typically are not ontologies. But when they are used to answer "What exists?" and thus when they formally capture the concepts, relationships, and constraints that exist in a domain, they are ontologies.

of a WoK, it can answer the question: "What are the known facts, both given and implied?" We ground our conceptual model in a description logic—a decidable fragment of first-order logic [BN03]. To make this logic practical for non-logicians, we must and do add a query generator whose input consists of ordinary free-form textual expressions or ordinary fill-in-the-blank query forms. Both query modes fundamentally depend on conceptual-model-based ontologies to convert free-form and form-based queries to structured queries. Justification of query results relies on tracing annotation links back to source data and on following reasoning chains.

To illustrate these ideas, we give some examples. Suppose we wish to find a used car to purchase. We might pose this query: "Find me a red Nissan for under $5000, a 1990 or newer with less than 100K miles on it", or this query: "I'd like a Japanese-made car for under 15 grand." Figure 15.1 shows some web pages with cars for sale that satisfy these queries. Two of the three Nissans satisfy the first query, and all the Nissans and the Mitsubishi, but not the Toyota, satisfy the second query. Unfortunately, however, search engines do not access the facts within these ads in the way we would wish to find these cars. Our approach of superimposing a web of knowledge over a web of pages makes these facts visible from outside the page and directly accessible to query engines (as opposed to search engines).

To make this work, we need an ontology for car ads. Figure 15.2 shows an example. The ontology is a conceptual model. It consists of object sets which are either lexical (dashed boxes in Figure 15.2) or non-lexical (solid boxes). Instances in lexical object sets are strings of characters such as 'Nissan' or '1990'; whereas instances in non-lexical object sets are object identifiers that stand for real-world objects—Car_{73} and Car_{1194}, for example, identify specific cars. Relationship sets in the conceptual model are lines connecting object sets. Min-max participation constraints impose restrictions on the relationship sets: a car described in a car ad can have zero or more features (*0:**), but at most one make (*0:1*). Ontologies for our WoK vision also support hypernym-hyponym is-a hierarchies and holonym-meronym part-of hierarchies. A white triangle denotes an is-a hierarchy, so that, for example, a *Body Type* is-a *Feature* in Figure 15.2. The plus symbol (+) in the triangle specifies a disjointness constraint among the specializations; it is also possible to declare union constraints ($\cup$) specifying that a generalization object set is a union of its specialization object sets and partition constraints ($\uplus$) specifying that specialization object sets partition their generalization. A black triangle denotes a part-of hierarchy. In Figure 15.2, for example, a model such as "Accord" aggregated with a trim specification such as "LX" constitutes the concept "Accord LX", which has some *ModelTrim* object identifier (e.g., $ModelTrim_{39}$).

Whereas an ontology tells us what kind of knowledge exists in our domain of interest, epistemological specifications tell us what the knowledge is and how it is acquired. For our WoK vision, populating an ontology yields knowledge. We can populate the the car-ad ontology in Figure 15.2


City Weekly
news | arts & entertainment | event listings | dining listings | classifieds | best of utah | about...
enter keywords here
search
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
OnlineAthens
Athens Banner-Herald
Bona Fide Classified
Athens, GA
NEWS | SPORTS | DOGBYTES | ROCKATHENS | CLASSIFIEDS | JOBS | HOMES | AUTOS | CITY GUIDE
Classifieds
Real Estate Sales
Real Estate For Rent
Employment
Financial
Transportation
Rec. Vehicles
Previous Page
Next Page
Clear My List
View My List
Add to My List

Place your ad in the *City Weekly* classifieds! 801-575-7028 | email

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 con

'99 PORSCHE
911, Carrera, mint condition, S
sacrifice $39,900. Call 209-21

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

March 14
397 vehicles
132 jobs
176 homes

TRANSPORTATION

(385 results) - Displaying 1-25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

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.
		TOYOTA TUNDRA	TOYOTA 4WD V8 2002, SR5 Tundra, regular cab. 8' bed. Loaded with upgrades. 100k warranty. Line-X. $19,800. 706-769-4323.
		HEVROLET 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.
		RD BRONCO	FORD BRONCO 1986, 302 engine, 4 wheel drive, 116k miles, good condition, runs good. $3,450 negotiable. Call 706-367-9061.
		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.

2003 Nissan Altima

Price: $6,990

Year:	2003
Make:	Nissan
Model:	Altima
Trim:	Base; S; SL
Engine:	4 Cylinder Gasoline
Trans:	Automatic
Fuel:	Gasoline
Color:	Red
Interior:	Black
Miles:	99853
VIN:	1N4AL11D73C115139
Stock #:	115139
Body Style:	Unspecified
Condition:	Used
Category:	Used Cars

Consumer Reviews
We found 2 reviews for 2003 Nissan Altima's
Read the reviews!

Fig. 15.1 Sample Car Ad Web Pages.

with facts derived from car ads. Assuming *Car*$_{73}$ denotes the Nissan Altima pictured in Figure 15.1, some of the object-set facts are: *Car(Car*$_{73}$*)*, *Year(2003)*, and *Transmission("Automatic")*, and some of the relationship facts are: *CarYear(Car*$_{73}$*, 2003)* and *CarFeature(Car*$_{73}$*, "Automatic")*.

The way we intend to acquire knowledge in our WoK vision is particularly interesting. Although possible to simply encode by hand, this is far too labor intensive and does not scale. We must find ways to automatically identify facts and associate them with ontologies. Sometimes information is structured in such a way that it is possible to reverse-engineer it into an ontology; sometimes it is possible to resort to available outside knowledge sources to align semi-structured information with an ontology; but sometimes the information yields to neither of these techniques. For this latter case, we augment the ontologies themselves so that they are capable of recognizing, annotating, and extracting relevant facts with respect to their ontological descriptions. We call these augmented ontologies *extraction ontologies*, because they are

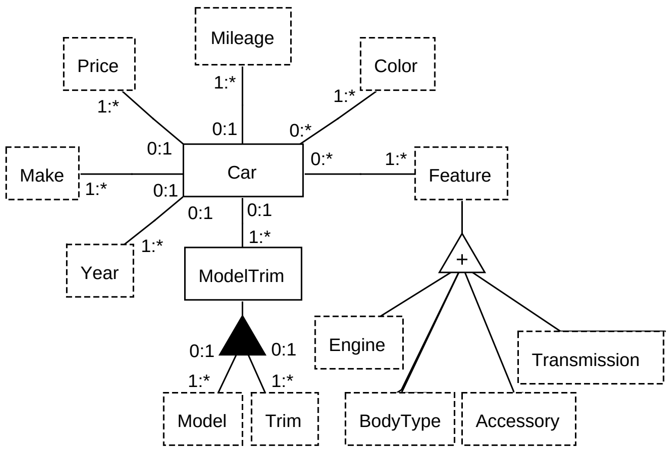


Fig. 15.2 Car Ad Ontology.

capable of locating and extracting facts from any kind of document, unstructured as well as semi-structured and structured.

The augmentation that turns an ontology into an extraction ontology is a data frame [Emb80]. Data frames are linguistically grounded abstract data types—they encapsulate everything we wish to know about categorized data instances including their internal and external representations and their applicable operations. The linguistic grounding consists of providing recognizers for instances of object and relationship sets. Figure 15.3 shows examples of the kind of recognizers we currently use in our WoK vision, but any kind of information extractor [Sar08] is possible: e.g., machine-learned wrappers (e.g., [WCW$^+$09]), NLP-based recognition techniques (e.g., [FM09]), and elaborate handcrafted rules (e.g., [GGTB08, HLF$^+$08]). The recognizers in Figure 15.3 are of two types—regular expressions and lexicons—which operate independently or together. The regular expression for the external representation of *Price* in Figure 15.3 recognizes price instances such as "$23,900" and "15 grand". The external-representation recognizer for *Make*, on the other hand, is a lexicon that lists all the makes of cars, including their alternate spellings and abbreviations. Context keywords such as *price* and *asking* in Figure 15.3 help disambiguate instances that may be recognized for more than one concept (e.g., MSRP price vs. asking price). Operators also have recognizers. Keywords such as "less than", "<", and "under" in Figure 15.3 indicate the applicability of the *LessThan* operator. A price instance following these keywords becomes the second parameter p_2 in the operation, the car price being assumed as the first.

Justification of captured knowledge is a natural consequence of acquiring and semantically annotating knowledge. When we reverse engineer structured

```
Price
    internal representation: Integer
    external representation: \$[1-9]\d{0,2},?\d{3} | \d?\d [Gg]rand | ...
    context keywords: price|asking|obo|neg(\.|otiable)| ...
    ...
    LessThan(p1: Price, p2: Price) returns (Boolean)
    context keywords: (less than | < | under | ...)\s*{p2} | ...
    ...
Make
    ...
    external representation: CarMake.lexicon
    ...
```

Fig. 15.3 Data Frames.

knowledge into a fact-filled ontology, align facts in a knowledge source with an ontology, or extract facts from data-rich documents, the WoK system keeps track of the source of each fact. Later, when someone queries for these facts, the WoK system provides, in addition to the standard query results, a cached page with the fact instances highlighted. For example, if we query for a red Nissan Altima for under 15 grand, as part of the answer the WoK system can retrieve the cached page of the Altima in Figure 15.1 and display it with the strings *Red*, *Nissan*, *Altima*, and *$6,990* highlighted.

Besides enabling fact recognition in source documents, extraction ontologies also enable free-form query processing. For example, a WoK system with the ontology in Figure 15.1 augmented with the data-frame recognizers in Figure 15.3 can interpret and process the query: “Find me a red Nissan Altima for under 15 grand.” The system associates the recognized instances “red”, “Nissan”, “Altima”, and “15 grand” respectively with the object sets *Color*, *Make*, *Model*, and *Price*, and it associates “under” with the *LessThan* operator in the *Price* data frame and the price “15 grand” with the operator’s parameter *p2*. Generating a formal select-project-join query from these recognized associations is straightforward: do outer join over the ontology’s structure; select based on the identified constants and Boolean operations; and project on the mentioned object sets.

Retrieving inferred facts is more complex. To illustrate, consider processing the query, “I’d like a Japanese-made car for under 15 grand.” For this query the WoK system needs a way to determine which cars are Japanese. Suppose we have a second ontology about car manufacturers such as the one in Figure 15.4. Observe that the concept $Make$ in Figure 15.1 is semantically the same as the concept $Make$ in Figure 15.4. Connecting these concepts with a between-ontologies *same-as* link is an example of the interconnecting links in the web-like structure constituting a WoK. With these two ontologies and the interconnecting link, we now have the information we need to declare inference rules that can reason about a car being Japanese-made. A car in a car ad that has a make produced by a manufacturer whose headquarters is in Japan is a Japanese-made car.

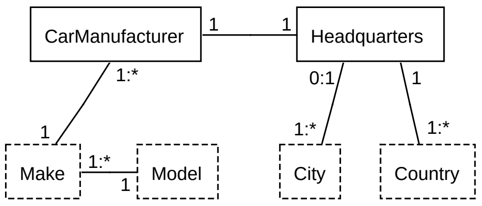


Fig. 15.4 Car Manufacturer Ontology.

Next observe that populated ontologies are in reality first-order-language theories of predicate calculus. Each object set S is a one place predicate $S(x)$. Each n-ary relationship set R is an n place predicate, $R(x_1, ..., x_n)$. The constraints of our conceptual-modeling language are all expressible as closed well-formed formulas of predicate calculus. Populating an ontology provides the ground facts for a first-order theory. When all the constraints hold for a populated ontology, we have a model of the first order theory. And, we can reason over the model with logic languages, appropriately restricted to make them decidable and tractable.

For our query about Japanese-made cars, suppose we have the following logic rule (expressed in Datalog-like syntax) for the car-manufacturer ontology:

```
JapaneseMake(x) :- Make(x), CarManufacturerMake(y, x),
        CarManufacturerHeadquarters(y, z),
        CountryHeadquarters('Japan', z).
```

Similar to linguistically grounding operators in our ontologies, we can linguistically ground logic rules. Thus, the phrase "Japanese-made car" should indicate that this rule applies. Now, when a user poses a query for Japanese-made cars for under 15 grand, the WoK system recognizes that the two ontologies apply and generates the following formal query over the two theories *CarAd* and *CarManuf*:

```
CarAd.Car(x) :- CarAd.CarPrice(x, y),
        CarAd.LessThan(y, 15000), CarAd.CarMake(x, z),
        CarManuf.JapaneseMake(z).
```

Here, the *same-as* link lets us seamlessly navigate among populated ontologies in a WoK. As an aside, note the conversion of "15 grand" to the integer 15000. As indicated in the *Price* data frame in Figure 15.3, the type for the internal representation is *integer*. As the WoK system extracts data for an ontology, it also converts the data to declared internal representations.

By way of summary for this informal introduction to our WoK vision, we see a WoK consisting of logic theories, interconnected and superimposed over web documents. Logic theories are populated ontologies. From an epistemological point of view, populated ontologies are extensional knowledge.

And with the addition of inference rules, populated ontologies also embody intensional knowledge.

15.3 WoK Formalization

We base our foundational conceptualization for a web of knowledge on the conceptual modeling language OSM (Object-oriented Systems Modeling) [EKW92]. OSM, however, simply provides a graphical representation of a first-order-logic language. Here we restrict OSM to be decidable, yet powerful enough to represent desired ontological concepts and constraints. We call our restriction *OSM-O*, short for *OSM-Ontology*. We thus base our foundational conceptualization directly on an appropriate restriction of first-order logic. This WoK foundation should be no surprise since it is the basis for modern information systems and has been the basis for formalizing information since the days of Aristotle [AriBC].

Definition 1. *OSM-O* is a triple (O, R, C):

- O is a set of object sets; each is a one-place predicate; and each predicate has a *lexical* or a *non-lexical* designation.
- R is a set of n-ary relationship sets ($n \geq 2$); each is an n-place predicate.
- C is a set of constraints:
  - Referential integrity: $\forall x_1 ... \forall x_n (R(x_1, ..., x_n) \Rightarrow S_1(x_1) \wedge ... \wedge S_n(x_n)$ for each n-ary relationship set R connecting object sets S_1, ..., S_n.
  - Participation constraint *min*:*max* cardinality: for every connection of an object set S to an n-ary relationship set R, $\forall x_i(S(x_i) \Rightarrow \exists^{\geq min}{<}x_1$, ..., x_{i-1}, x_{i+1}, ..., $x_n{>}(R(x_1, ..., x_n)))$ if $min > 0$, and $\forall x_i(S(x_i) \Rightarrow \exists^{\leq max}{<}x_1$, ..., x_{i-1}, x_{i+1}, ..., $x_n{>}(R(x_1, ..., x_n)))$ if max is not * (the symbol denoting an unbounded maximum).
  - Generalization/specialization: $\forall x(S_1(x) \vee ... \vee S_n(x) \Rightarrow G(x))$ for each generalization object set G of specialization object sets S_1, ..., S_n in a hypernym-hyponym is-a hierarchy. In addition, $\forall x(S_i(x) \Rightarrow \neg S_j(x))$ for $1 \leq i,j \leq n$ and $i \neq j$ if the specialization object sets are disjoint and $\forall x(G(x) \Rightarrow S_1(x) \vee ... \vee S_n(x))$ if the generalization object set is complete—is a union of the specialization object sets.
  - Aggregation: holonym-meronym relationship sets grouped as an aggregation in an is-part-of hierarchy. □

Example 1. Figure 15.2 shows an OSM-O model instance. $Car(x)$ and $Model$-$Trim(x)$ are the one-place predicates for the two non-lexical object sets. $Mileage(x)$ and $Engine(x)$ are two of the one-place predicates for the lexical object sets. $CarYear(x, y)$ is a two-place predicate for the relationship set connecting Car and $Year$. For readability, we may provide a more descriptive name for a relationship set so long as the naming phrase for the relationship set includes the names of its object sets. We then typically use infix notation and write, for example, $Car(x)hasYear(y)$ for the $CarYear(x, y)$ relationship set or $Car(x)costsPrice(y)$ for the $CarPrice(x, y)$ relationship set. One of the referential integrity constraints is $\forall x \forall y(CarYear(x, y) \Rightarrow Car(x) \wedge Year(y))$. One of the participation constraints is $\forall x(Car(x) \Rightarrow \exists^{\leq 1} y(CarYear(x, y)))$, where we drop the tuple-grouping angle brackets for the common case of only one variable being existentially quantified. The formula $\forall x(Engine(x) \vee BodyType(x) \vee Accessory(x) \vee Transmission(x) \Rightarrow Feature(x))$ defines the generalization/specialization; its disjointness constraint includes $\forall x(Engine(x) \Rightarrow \neg\ BodyType(x))$ as one of its terms. An aggregation groups several relationship sets denoting subparts of superparts: $Model(x)isSubpartOfModelTrim(y)$ and $Trim(x)isSubpartOf$-$ModelTrim(y)$ are the two relationship sets of the aggregation in Figure 15.2; their inverses are respectively $ModelTrim(x)isSuperpartOfTrim(y)$ and $ModelTrim(x)isSuperpartOfModel(y)$. Although graphical in appearance, an OSM-O diagram is merely a two-dimensional rendition of predicates and closed formulas as defined in Definition 1. □

Definition 2. Let $M = (O, R, C)$ be an OSM-O model instance. Let I be an interpretation for M that has a domain $D = L_{ID} \cup O_{ID}$, where $L_{ID} \cap O_{ID} = \emptyset$, and a declaration of **True** or **False** for each valid instantiation of each predicate in $O \cup R$. For predicates in O, valid instantiations require lexical predicates to be instantiated with values in L_{ID} and non-lexical predicates to be instantiated with values in O_{ID}. For predicates in R, valid instantiations require each value v to be lexical or non-lexical according to whether the connected object set for v is lexical or non-lexical respectively. If all the constraints of C hold, I is a *model* of M, which we call a *valid interpretation* of M (to avoid an ambiguous use of the word "model" when also discussing conceptual models). An instantiated **True** predicate for a valid interpretation is a *fact.* □

Example 2. A valid interpretation of the OSM-O model instance in Figure 15.2 contains facts about cars. A valid interpretation might include the facts $Car(Car_3)$, $Year(2003)$, $CarYear(Car_3, 2003)$, $Model(\text{"Accord"})$, $Trim(\text{"LX"})$, $ModelTrim(ModelTrim_{17})$, $CarModelTrim(Car_3, ModelTrim_{17})$, and $Trim(\text{"}LX\text{"})isPartOfModelTrim(ModelTrim_{17})$. The object sets Car and $ModelTrim$, being non-lexical, have object identifiers for their domain-value substitutions (which we denote by object-set names with a subscript). The constraint $\forall x(Car(x) \Rightarrow \exists^{\leq 1} y(CarYear(x, y)))$ holds for $Car(Car_3)$ if $CarYear(Car_3, 2003)$ is the only car-year pair that exists with Car_3 as its first element. □

Similar to the work by Buitelaar, et al. [BCHS09], we now show how to linguistically ground OSM-O. Linguistically grounding OSM-O turns OSM-O model instances into *OSM-Extraction-Ontology* model instances (*OSM-EO* model instances). We begin by defining an ordinary abstract data type for each object set and relationship set. We then add linguistic recognizers for instance values, operators, operator parameters, and relationships.

Definition 3. An *abstract data type* is a pair (V, O) where V is a set of values and O is a set of operations. □

Definition 4. A *data frame* is an abstract data type augmented as follows:

1. The data frame has a name N designating the set of values V, and it may have a list of synonyms for N.
2. The value set V has instance recognizers that identify lexical patterns denoting values in V.
3. For a lexical object set, the operator set O includes input operators to convert identified instances to an internal representation and output operators to convert the internal representation of instances to displayable strings.
4. An operation o in O may have a recognizer that identifies lexical patterns in text that indicate that o applies. Further, the recognizer identifies lexical patterns that, along with instance recognizers, identify parameters for o. □

Example 3. In Figure 15.3 the value set V for the *Price* data frame is of type *Integer*. Its recognizer is a potentially lengthy list of regular expressions augmented by keywords. Its operation set O includes the *LessThan* operator and potentially has many more operations. The *LessThan* operator has keyword phrases that indicate its applicability as well as how to identify its parameters. □

For relationship sets, the definition of a data frame does not change, but a typical view of the definition shifts as we allow value sets to be n-tuples of values rather than scalar values. Further, like recognizers for operators, they rely on instance recognizers from the data frames of their connected object sets.

Example 4. Suppose the Car object set in Figure 15.2 has a relationship set to a $Person$ object set. The relationship-set data frame may have recognizers for any one of several possible relationships such as $\{Person\}$ *is selling* $\{Car\}$, $\{Person\}$ *posted* $\{Car\}$ *ad*, or $\{Person\}$ *is inquiring about* $\{Car\}$ Here, the braces enclose references to data frames for the non-lexical object sets *Car* and *Person.* □

As is standard, implementations of abstract data types are hidden, and we hide implementations for data frames as well. Similar to data independence in database systems, this approach accommodates any implementation. In particular, it allows for new and better recognizers, which we can draw from the large body of work devoted to information extraction [Sar08].

Definition 5. If M is an OSM-O model instance with a data frame for each object set and relationship set, M is an *OSM-EO* model instance. □

An OSM-EO model instance is linguistically grounded in the sense that it can both "read" and "write" in some natural language. To "read" means to be able to recognize facts in natural language text and to extract fact instances with respect to the ontology in the OSM-EO model instance. To "write" means to display fact instances so that they are human-readable.

How well a particular OSM-EO model instance can "read" and "write" makes a difference in how well it performs. Our experience is that OSM-EO model instances can "read" some documents well (over 95% precision and recall [ECJ+99]), but it is clear that opportunities abound for further research and development. Writing human-understandable descriptions is less difficult to achieve—just select any one of the phrases for each object set and relationship set (e.g., $Person(Person_{17})$ *is selling* $Car(Car_{734})$, $Car(Car_{734})$ *has* $Make(Honda)$). Making written descriptions more pleasing, of course, is more difficult.

Continuing in our quest to define the components of our WoK vision, we now define its "knowledge," and we explain how we see its "knowledge" being justified and how we envision its knowledge components being interconnected.

Definition 6. The collection of facts in an OSM-O model instance constitutes its *extensional knowledge*. The collection of implied facts derived from the extensional knowledge by inference rules[2] constitutes its *intensional knowledge*. The extensional and intensional knowledge together constitute the *knowledge* of the OSM-O model instance. □

Although this view of knowledge is common in computing, Plato, and those who follow his line of thought, also demand of knowledge that it be a "justified true belief" [PlaBC]. "Knowledge" without some sort of truth authentication can be unsupported and even misleading. For our vision of a WoK, we attempt to establish truth via provenance and authentication. When an extraction ontology extracts a fact from a source document, it retains a link to the fact; and when a query answer requires reasoning over rules, the system records the reasoning chain. Users can ask to see fact sources and rule chains, and in this way they can authenticate facts and reasoning the way we usually do—by checking sources and fact-derivation rules.

Definition 7. A *knowledge bundle* (*KB*) is a 5-tuple (O, E, S, I, R) where O is an OSM-O model instance; E is an OSM-EO instance whose OSM-O instance is O; S is a set of source documents from which facts for E are extracted; I is a valid interpretation for O whose facts are extracted from the documents in S; and R is a set of inference rules. □

Finally, to make the envisioned WoK truly a *web* of knowledge, we interconnect knowledge bundles (KBs). Facts about the same object may appear

[2] As the work on logic and particularly on description logics [BN03] continues to expand, we can take advantage of the work of this community (e.g., [Ros05, CGL09, CGP09]) to employ better and more powerful reasoning engines.

in more than one KB. We can directly connect these objects so that users may navigate among KBs and obtain additional information about an object. Concepts in more than one KB may also be essentially the same as is the concept $Make$, and also the concept $Model$, in their respective ontologies in Figures 15.2 and 15.4. We also connect concepts across ontologies to provide additional navigation paths.

Definition 8. A *Web of Knowledge* (*WoK*) is a collection of knowledge bundles interconnected with binary links, $<x, y>$, of two types: (1) *object identity:* non-lexical object identifier x in knowledge bundle B_1 refers to the same real-world object as non-lexical object identifier y in knowledge bundle B_2. (2) *Object-set identity:* object set x in knowledge bundle B_1 designates the same kind of real-world objects as object set y in knowledge bundle B_2. □

15.4 WoK Construction

To construct a WoK, we must be able to construct a knowledge bundle (KB), and we must be able to establish links among KBs. We can construct KBs and establish links among them by hand (and this should always be an option). However, scaling WoK construction demands semi-automatic procedures, with much of the construction burden placed on the system—all of it when possible. Our KB construction tools transform, or aid in transforming, source information into KB components. For links among KBs we apply record-linkage and schema-mapping tools.

Definition 9. A *transformation* is a 4-tuple (R, S, T, Σ), where R is a set of resources, S is the source conceptualization, T is the target conceptualization for an S-to-T transformation, Σ is a set of source-to-target transformation statements. □

Definition 9 leaves several of its components open—to take on specific meanings in a variety of KB building tools. The "set of resources" is undefined, but we intend this to mean semantic resources such as WordNet and a data-frame library. "Target conceptualizations" are KBs or KB components. "Source conceptualizations" depend on sources whose fact conceptualizations can be formal, semi-formal, or informal. "Source-to-target transformation statements" can be declarative or procedural and can be written in a variety of formal languages.

To the extent possible, we want our transformations to preserve the information and constraints in source documents and repositories. When sources are formalized as predicate calculus or in a formalization equivalent to predicate calculus, we can guarantee the preservation information and constraints. We identify the predicates and the facts for the predicates (thus preserving information) and formulate a closed well-formed formula for each constraint (thus preserving constraints). If the source interpretation is valid, the target interpretation will be valid as well. When sources are informal with respect to predicate calculus, the predicates, facts for the predicates, and the constraints are implicit. The challenge is nevertheless to discover and extract them.

Definition 10. Let S be a predicate calculus theory with a valid interpretation, and let T be a populated OSM-O model instance constructed from S by a transformation t. Transformation *t preserves information* if there exists a procedure to compute S from T. Let C_S be the closed, well-formed formulas of S, and let C_T be the closed, well-formed formulas of T. Transformation t *preserves constraints* if $C_T \Rightarrow C_S$. □

Our goal has been and is to successfully develop automatic and good semi-automatic transformations over a broad spectrum of documents for a variety of ontological contexts. For sources whose facts and constraints over these facts have formal declarations, transformations should preserve all facts and constraints. For sources whose facts and constraints are implicit, we seek to identify the facts and constraints that are applicable to a given ontology or, in the absence of a given ontology, we seek to determine and populate the implicit ontology based on the document's data and on applicable external knowledge resources.

Longstanding research endeavors can all contribute to various parts of WoK construction. These include: reverse engineering [Aik98], table and form understanding [EHLN06, dSJT06], ontology learning [Cim06], ontology alignment [ES07], data integration [BN08, HRO06, RB01], and record linkage [EIV07]. In Sections 15.4.1–15.4.4 we explain how we have taken advantage of some of this work for WoK construction. In Section 15.4.1 we show how to reverse engineer XML data repositories into KBs. In Section 15.4.2 we describe how we can interpret collections of nested tables in hidden web pages and thus turn the collection into a KB. In Section 15.4.3 we explain how we integrate a group of semantically overlapping tables to create a KB. And, in Section 15.4.4 we give a way via form creation and information harvesting to generate KBs. Finding and implementing other ways to construct WoK components are interesting and worthwhile research endeavors.

15.4.1 Construction via XML Reverse Engineering

We have designed a conceptual model for XML, called C-XML (Conceptual XML). C-XML adds a few XML-specific concepts to OSM-O, including in particular a sequencing construct and a choice construct. Being formally defined as templates over OSM-O constructs, however, these additions remain within the purview of OSM-O.

Reverse engineering an XML schema to C-XML effectively defines a mapping to OSM-O for all XML documents complying to the XML schema [AK07]. The basic translation strategies for mapping XML Schema to C-XML are straightforward, although some parts of the translation require some sophisticated manipulation. In the translation, elements and attributes become object sets. Elements that have simple types become lexical object sets, while elements that have complex types become non-lexical object sets. Attributes become lexical object sets since they always have a simple type. Built-in data types and simple data types for an element or an attribute in XML Schema are specified in the data frame associated with the object set representing the element or the attribute. XML parent-child connections among elements and XML element-attribute connections both become binary relationship sets in C-XML. The constraints *minOccurs* and *maxOccurs* translate directly to participation constraints in C-XML.

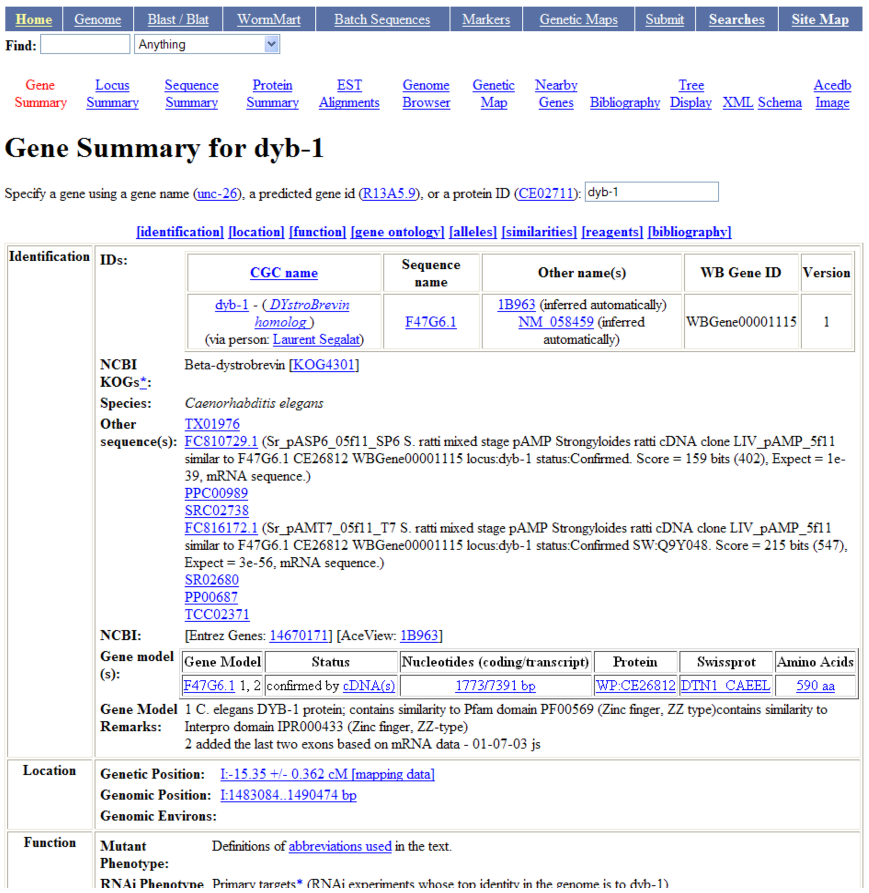
Unfortunately, not everything is straightforward. Translations for keys, extension, restriction, substitution groups, and mixed content are all quite interesting. The translation also involves a myriad of detail extending to over 40 pages in [AK07]. Although extensive, the translation details provide a constructive proof that the transformation from XML Schema to C-XML preserves both information and constraints.

The result of doing information- and constraint-preserving transformations of XML documents complying to an XML schema is a KB. Further, to the extent that we can automatically infer an XML schema specification directly from an XML document, we can also reverse-engineer raw XML documents into populated C-XML model instances and thus into KBs.

15.4.2 Construction via Nested Table Interpretation

TISP (Table Interpretation with Sibling Tables) is a tool of ours that interprets tables in sibling pages [Tao08]. To interpret a table is to properly associate table category labels with table data values. Using Figure 15.5 as an example, we see that *Identification*, *Location*, and *Function* are labels for the large rectangular table. Inside the cell labeled *Identification* is another table with headers *IDs*, *NCBI KOGs*, *Species*, etc. Nested inside of this table are two more tables, the first starting with the label *CGC name* and the second starting with the label *Gene Model*. We associate labels with data values by observing the table structure. A cell in a table associates with its header label (or labels in the case of multi-dimensional tables). For nested tables, we trace the sequence of labels from the outermost label to the data cell. Thus, for example, the label for the value *F47G6.1* under *Sequence name* is *Identification.IDs.Sequence_name*.

Although automatic table interpretation can be complex, if we have another page, such as the one in Figure 15.6, that has essentially the same


Home | Genome | Blast / Blat | WormMart | Batch Sequences | Markers | Genetic Maps | Submit | Searches | Site Map

Find: Anything

Gene Summary | Locus Summary | Sequence Summary | Protein Summary | EST Alignments | Genome Browser | Genetic Map | Nearby Genes | Bibliography | Tree Display | XML Schema | Acedb Image

Gene Summary for dyb-1

Specify a gene using a gene name (unc-26), a predicted gene id (R13A5.9), or a protein ID (CE02711): dyb-1

[identification] [location] [function] [gene ontology] [alleles] [similarities] [reagents] [bibliography]

Identification

IDs:

CGC name	Sequence name	Other name(s)	WB Gene ID	Version
dyb-1 - (*DYstroBrevin homolog*) (via person: Laurent Segalat)	F47G6.1	1B963 (inferred automatically) NM_058459 (inferred automatically)	WBGene00001115	1

NCBI KOGs*: Beta-dystrobrevin [KOG4301]

Species: *Caenorhabditis elegans*

Other sequence(s):
TX01976
FC810729.1 (Sr_pASP6_05f11_SP6 S. ratti mixed stage pAMP Strongyloides ratti cDNA clone LIV_pAMP_5f11 similar to F47G6.1 CE26812 WBGene00001115 locus:dyb-1 status:Confirmed. Score = 159 bits (402), Expect = 1e-39, mRNA sequence.)
PPC00989
SRC02738
FC816172.1 (Sr_pAMT7_05f11_T7 S. ratti mixed stage pAMP Strongyloides ratti cDNA clone LIV_pAMP_5f11 similar to F47G6.1 CE26812 WBGene00001115 locus:dyb-1 status:Confirmed SW:Q9Y048. Score = 215 bits (547), Expect = 3e-56, mRNA sequence.)
SR02680
PP00687
TCC02371

NCBI: [Entrez Genes: 14670171] [AceView: 1B963]

Gene model (s):

Gene Model	Status	Nucleotides (coding/transcript)	Protein	Swissprot	Amino Acids
F47G6.1 1, 2	confirmed by cDNA(s)	1773/7391 bp	WP:CE26812	DTN1_CAEEL	590 aa

Gene Model Remarks: 1 C. elegans DYB-1 protein; contains similarity to Pfam domain PF00569 (Zinc finger, ZZ type)contains similarity to Interpro domain IPR000433 (Zinc finger, ZZ-type)
2 added the last two exons based on mRNA data - 01-07-03 js

Location

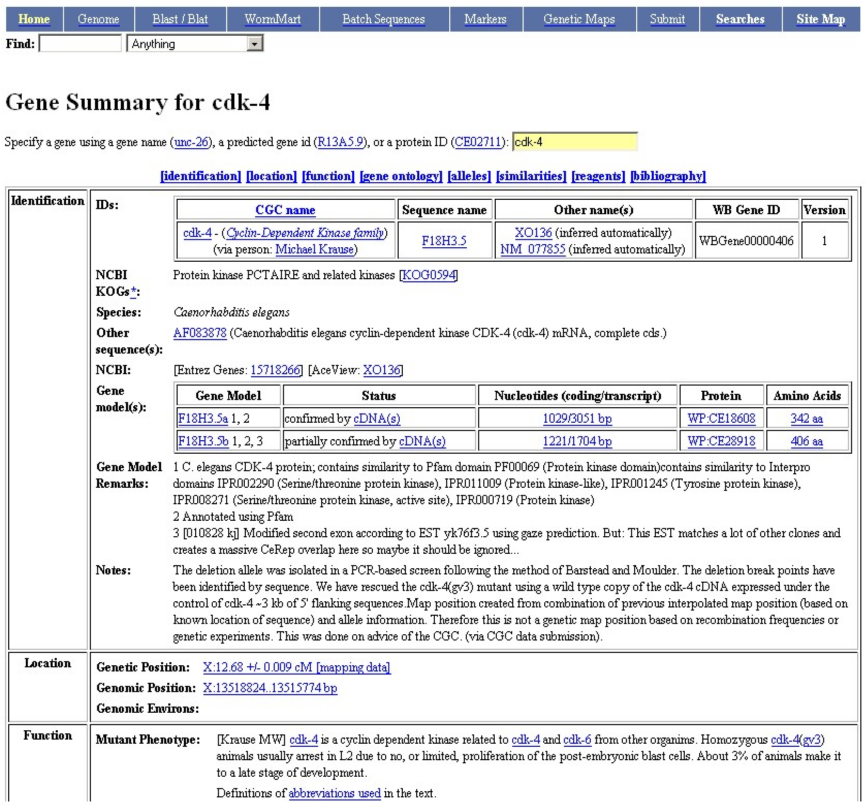
Genetic Position: I:-15.35 +/- 0.362 cM [mapping data]
Genomic Position: I:1483084..1490474 bp
Genomic Environs:

Function

Mutant Phenotype: Definitions of abbreviations used in the text.
RNAi Phenotype Primary targets* (RNAi experiments whose top identity in the genome is to dyb-1)

Fig. 15.5 Nested Table from a Molecular-Biology Web Page.

structure, the system can usually obtain enough information to make automatic interpretation possible. We call pages that are from the same web site and have similar structures *sibling pages*. The two pages in Figures 15.5 and 15.6 are sibling pages. They have the same basic structure, with the same top banners that appear in all the pages from this web site, with the same table title (*Gene Summary for* some particular gene), and a table that contains information about the gene. Corresponding tables in sibling pages are called *sibling tables*. If we compare the two large tables in the main part of the sibling pages, we can see that the first columns of each table are exactly the same. If we look at the cells under the *Identification* label in the two tables, both contain another table with two columns. In both cases, the first column contains the identical labels *IDs*, ..., *Remarks*, although the table in Figure 15.6 has one additional label, *Notes*. Further, the tables under *Identification.IDs* also have identical header rows, and the tables under *Iden-*



Gene Summary for cdk-4

Specify a gene using a gene name (unc-26), a predicted gene id (R13A5.9), or a protein ID (CE02711): cdk-4

[identification] [location] [function] [gene ontology] [alleles] [similarities] [reagents] [bibliography]

Identification

IDs:

CGC name	Sequence name	Other name(s)	WB Gene ID	Version
cdk-4 - (*Cyclin-Dependent Kinase family*) (via person: Michael Krause)	F18H3.5	XO136 (inferred automatically) NM_077855 (inferred automatically)	WBGene00000406	1

NCBI KOGs*: Protein kinase PCTAIRE and related kinases [KOG0594]

Species: *Caenorhabditis elegans*

Other sequence(s): AF083878 (Caenorhabditis elegans cyclin-dependent kinase CDK-4 (cdk-4) mRNA, complete cds.)

NCBI: [Entrez Genes: 15718266] [AceView: XO136]

Gene model(s):

Gene Model	Status	Nucleotides (coding/transcript)	Protein	Amino Acids
F18H3.5a 1, 2	confirmed by cDNA(s)	1029/3051 bp	WP:CE18608	342 aa
F18H3.5b 1, 2, 3	partially confirmed by cDNA(s)	1221/1704 bp	WP:CE28918	406 aa

Gene Model Remarks: 1 C. elegans CDK-4 protein; contains similarity to Pfam domain PF00069 (Protein kinase domain)contains similarity to Interpro domains IPR002290 (Serine/threonine protein kinase), IPR011009 (Protein kinase-like), IPR001245 (Tyrosine protein kinase), IPR008271 (Serine/threonine protein kinase, active site), IPR000719 (Protein kinase)
2 Annotated using Pfam
3 [010828 kj] Modified second exon according to EST yk76f3.5 using gaze prediction. But: This EST matches a lot of other clones and creates a massive CeRep overlap here so maybe it should be ignored...

Notes: The deletion allele was isolated in a PCR-based screen following the method of Barstead and Moulder. The deletion break points have been identified by sequence. We have rescued the cdk-4(gv3) mutant using a wild type copy of the cdk-4 cDNA expressed under the control of cdk-4 ~3 kb of 5' flanking sequences.Map position created from combination of previous interpolated map position (based on known location of sequence) and allele information. Therefore this is not a genetic map position based on recombination frequencies or genetic experiments. This was done on advice of the CGC. (via CGC data submission).

Location

Genetic Position: X:12.68 +/- 0.009 cM [mapping data]
Genomic Position: X:13518824..13515774 bp
Genomic Environs:

Function

Mutant Phenotype: [Krause MW] cdk-4 is a cyclin dependent kinase related to cdk-4 and cdk-6 from other organims. Homozygous cdk-4(gv3) animals usually arrest in L2 due to no, or limited, proliferation of the post-embryonic blast cells. About 3% of animals make it to a late stage of development.
Definitions of abbreviations used in the text.

Fig. 15.6 Sibling Page of the Page in Figure 15.5.

tification.Gene model(s) have nearly identical header rows. The data values, however, vary considerably. Generally speaking, we can look for commonalities in sibling tables to find labels and look for variations to find data values.

Given that we can interpret a table—find labels and values and properly associate them—we can create a conceptualization of the table linking labels as metadata with values as instance data. This simple conceptualization may not always be best, but for some tables it works well. In particular, it works well for nested tables like the ones in Figures 15.5 and 15.6.

Observe that for these nested tables, the conceptual, nested, label-value structure is isomorphic to a simple XML schema. There exists a single, nested, label path to every data value. For the sequence name *F47G6.1* in the top row of the table in Figure 15.5, the label path is *Identification.IDs.Sequence_name*. This nested-label property lets us conceptualize these tables in an XML-like conceptual tree with labels as tags and instance values as leaf strings in a nested tag structure, as Figure 15.7 illustrates. Note that the tree structure in Figure 15.7 precisely captures the nesting of the tables in Figures 15.5

and 15.6. Note also that the conceptualization can account for the variation among tables. In Figure 15.7, for example, *Gene Models* relates optionally to *Swissprot* because the ontology-generation process observes that one table (Figure 15.5) has an entry for *Swissprot* whereas the other (Figure 15.6) does not. The end result is an automatically generated KB containing all the data from the given set of sibling tables.

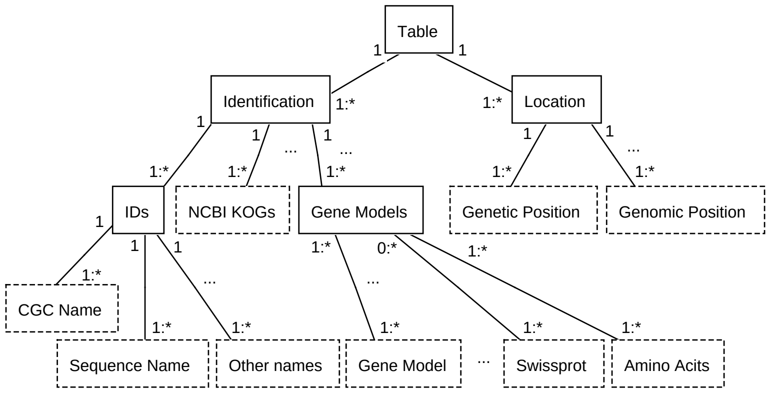


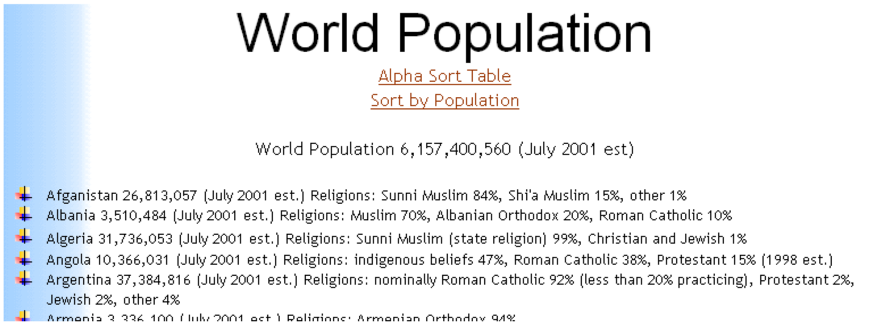
Fig. 15.7 Generated Conceptual Model Instance.

15.4.3 Construction via Semantic Integration

TANGO (Table ANalysis for Generating Ontologies) provides another way to create KBs. Given a collection of tables all in the same application domain, we reverse-engineer each table into a conceptual-model instance and then integrate the conceptual-model instances into an ontology that represents the domain. During the process, we analyze each table individually, inferring concepts, relationships among concepts, and data values for concepts and relationships. The result of this process is a conceptual-model instance for the table, which we call a "mini-ontology"—"mini" because the number of concepts in a table is usually small. We then exploit schema-mapping techniques to discover interrelationships among the mini-ontologies, enabling us to merge the generated conceptual-model instances into an ontological structure for the domain.

TANGO operates in three steps:

1. Recognize and canonicalize table information.
2. Construct mini-ontologies from canonicalized tables.


World Population

Alpha Sort Table

Sort by Population

World Population 6,157,400,560 (July 2001 est)

- Afganistan 26,813,057 (July 2001 est.) Religions: Sunni Muslim 84%, Shi'a Muslim 15%, other 1%
- Albania 3,510,484 (July 2001 est.) Religions: Muslim 70%, Albanian Orthodox 20%, Roman Catholic 10%
- Algeria 31,736,053 (July 2001 est.) Religions: Sunni Muslim (state religion) 99%, Christian and Jewish 1%
- Angola 10,366,031 (July 2001 est.) Religions: indigenous beliefs 47%, Roman Catholic 38%, Protestant 15% (1998 est.)
- Argentina 37,384,816 (July 2001 est.) Religions: nominally Roman Catholic 92% (less than 20% practicing), Protestant 2%, Jewish 2%, other 4%
- Armenia 3,336,100 (July 2001 est.) Religions: Armenian Orthodox 94%

Fig. 15.8 World Populations and Religions.

3. Discover inter-ontology mappings and merge mini-ontologies into a growing application ontology.

We illustrate with an example.

15.4.3.1 Table Recognition and Canonicalization

Tables appear in many shapes and sizes; most, but not all, have rectangular grid layouts. Figure 15.8 is an example of a table without a clearly delineated grid layout.

In our first step, we canonicalize tables by converting them to "Wang notation," a layout-independent formalization for tables [Wan96]. Wang notatation has two parts: (1) label structure and (2) data-value/label-structure association. Label structure consists of a collection of dimension trees, one for each coordinate that indexes a data cell. A dimension tree organizes labels in a tree structure: each path from root to leaf provides an index coordinate for a data value. In our example, the table in Figure 15.8 has two dimensions. In Wang notation, these two dimension trees are:

```
(DT1Root, {
            (Population (July 2001 est.), Ø),
            (Religion, {
                    (Albanian Orthodox, Ø),
                    (Muslim, Ø),
                    (Roman Catholic, Ø),
                    (Shi’a Muslim, Ø),
                    (Sunni Muslim, Ø),
                    ...
                    (other, Ø)
                    }))
(Country, {
            (Afghanistan, Ø),
            (Albania, Ø),
            ...
            })
```

Wang associates data with dimension trees in δ-statements. Each combination of paths through dimension trees can have a value:

δ(DT1Root.Population (July 2001 est.), Country.Afghanistan) = 26,813,057
δ(DT1Root.Religion.Albanian Orthodox, Country.Afghanistan) = $\perp$
...
δ(DT1Root.Religion.other, Country.Afghanistan) = 1%
δ(DT1Root.Population (July 2001 est.), Country.Albania) = 3,510,484
...

Here, the first statement is for the data cell containing 26,813,057, the population of Afghanistan. The other three index an empty cell, the 1% for “other” religions in Afghanistan, and the 3,510,484 for the population of Albania. Similarly, we can index all values for all countries in Figure 15.8.

We consider any collection of data that we can represent in Wang notation to be a table. Further, given a table in Wang notation, we can display the table in a standard grid form. We place the first dimension above the data, the second to the left of the data, the third to the left of the second with the second replicated for every leaf of the third, ... We omit implicit roots, such as the root *DT1Root* for Dimension Tree 1. Figure 15.9 displays the table in Figure 15.8 in this standard way. Figures 15.10–15.14 show several additional examples, which together with Figure 15.8 constitute, for our example here, the tables to be merged into a KB.

We [EHLN06, JNS+09, TE09], and others (e.g., [GBH+07, PSC+07, RK06, dSJT06, ZBC04]), are working toward fully automatic table-interpretation tools. These tools take as input a table such as the one in Figure 15.8, and produce as output Wang notation, which we can display in a standard way. In our tools, we augment Wang notation so that it can capture more than just labels and values. We also capture a table’s title, its footnotes, and its units of measure. In the absence of fully automated tools, we have developed

Country	Population (July 2001 est.)	Religion						
		Albanian Orthodox	Muslim	Roman Catholic	Shi'a Muslim	Sunni Muslim	...	other
Afghanistan	26,813,057				15%	84%		1%
Albania	3,510,484	20%	70%	10%				
...								

Fig. 15.9 Canonicalized Table for World Religious Populations.

Country	Population (July 2003 est.)	Median Age (2002)			Population Growth Rate (2003 est.)
		Total	Male	Female	
Afghanistan	28,717,213	18.9 years	19.1 years	18.7 years	3.38%*
Albania	3,582,205	26.5 years	24.8 years	28.1 years	1.03%
...					

* Note: this rate does not take into consideration the recent war and its continuing impact

Fig. 15.10 Canonicalized Table for People.

Country	Location Description	Geographic Coordinates
Afghanistan	Southern Asia, north and west of Pakistan, east of Iran	33 00 N, 65 00 E
Albania	Southeastern Europe, bordering on the Adriatic Sea and Ionian Sea, between Greece and Serbia and Montenegro	41 00 N, 20 00 E
...		

Fig. 15.11 Canonicalized Table for Geography.

tools that let a user efficiently mark a table's label areas, data areas, title, and other augmentations [JN08, PJK$^+$09]. As a principle, it should always be possible for a knowledge worker to manually specify any output to be generated by the system, even though we aim to automate as much as is possible. Manual specification ensures that we can always complete a task and that we can correct any errors our automated procedures may introduce.

15.4.3.2 Construction of Mini-Ontologies

Figure 15.15 gives a graphical representation of each of the mini-ontologies for our six sample canonicalized tables in Figures 15.9–15.14. The notation differs slightly from our earlier notation for OSM-O. In this notation, we represent graphically the four common participation constraints: *0:**, *0:1*, *1:**, and *1:1*. A zero minimum makes participation optional, which we denote with an "o" on the relationship-set line near the object set whose objects participate optionally. The absence of an "o" makes the participation

	Population
Asia	3,674,000,000
Africa	778,000,000
...	
New York City, New York	8,040,000
Los Angeles, California	3,700,000
...	
Mumbai, India	12,150,000
Buenos Aires, Argentina	11,960,000
...	
China	1,256,167,701*
India	1,017,645,163*
...	

*January 15, 2000

Fig. 15.12 Canonicalized Table for Largest Populations.

Place	Type	Elevation*	USGS Quad	Lat	Lon
Bonnie Lake	reservoir	unknown	Seivern	33 72 N	81 42 W
Bonnie Lake	lake	unknown	Mirror Lake	40 71 N	110 88 W
...					
New York	town/city	unknown	Jersey City	40 71 N	74 01 W
New York	town/city	149 meters	Leagueville	32 17 N	95 67 W
New York	mine	unknown	Heber City	40 62 N	111 49 W
...					

*Elevation values in this table are approximate, and often subject to a large degree of error. If in doubt, check the actual value on the map.

Fig. 15.13 Canonicalized Table for US Topographical Maps.

Pos	Language	Speakers	Where Spoken (Major)
1	Mandarin	885,000,000	China, Malaysia, Taiwan
2	Spanish	332,000,000	South America, Central America, Spain
3	English	322,000,000	USA, UK, Australia, Canada, New Zealand
...			

Fig. 15.14 Canonicalized Table for Most Spoken Languages.

mandatory—equivalent to a *1-minumum* in a participation constraint. Thus, for example, the mini-ontology in Figure 15(e) declares that a *Place* must have a *Name* and may, but need not, have an *Elevation*. A *1-maximum* in a participation constraint makes the relationship set functional, which we denote with an arrowhead on the opposite side of a relationship-set line. Thus, for example, in Figure 15(e), a *Place* has one *Name*, at most one *Elevation*, one *USGS Quad* (the map in which the center of the place appears), and one pair of *Geographic Coordinates*. The functional (arrowhead) notation, also allows us to express functional dependencies whose left-hand side is composite. Thus, for example, in Figure 15(a) we have the functional dependency

Country Religion → *Percent*. The notation also provides explicitly for object values with large black dots, which are object sets (one-place predicates) limited to have a single value.

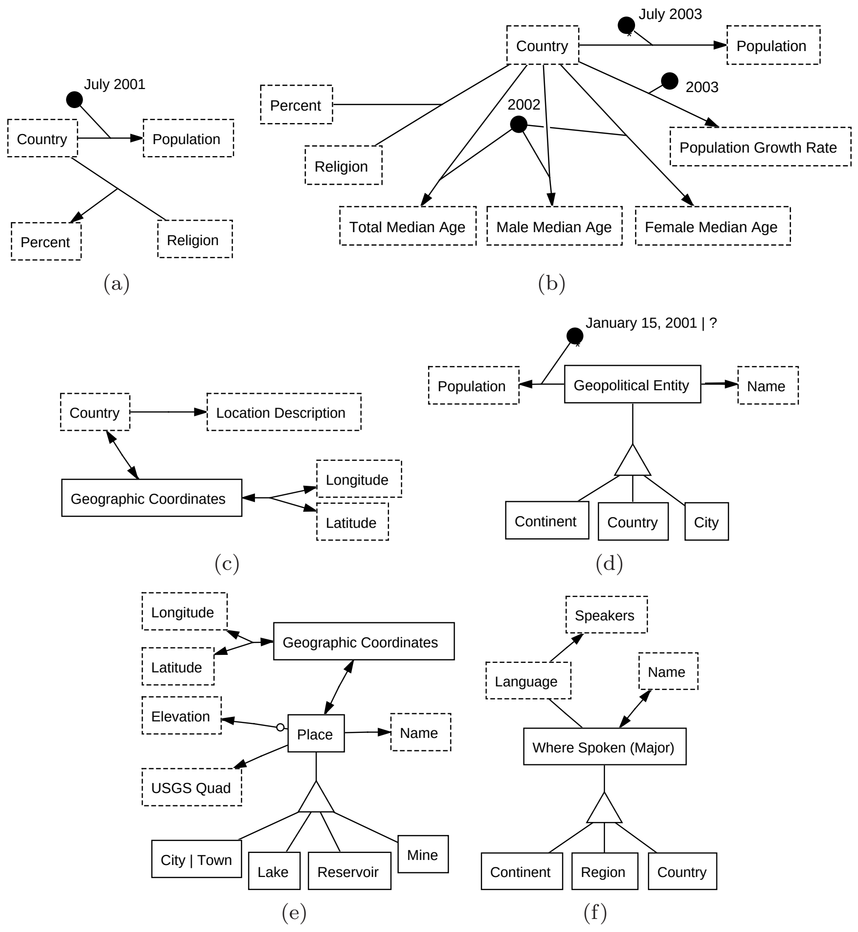


Fig. 15.15 Mini-Ontologies Constructed from the Tables in Figures 15.9 - 15.14.

To construct mini-ontologies from canonicalized tables, we must discover what concepts (object sets) are involved and how they are related (relationship sets). We must also determine the constraints that hold over the relationship sets (functional, mandatory/optional participation, aggregations) and among the object sets (generalization/specialization). We do so by appealing to the structural constraints of canonicalized tables and to outside resources such as WordNet and a data-frame library [LE09]. Aligning model instances with outside resources also makes them easier to integrate.

As an example, we obtain the mini-ontology in Figure 15(a) from the table in Figure 15.9 as follows. *Country* is a key and appears in a leftmost column, strongly suggesting that it should be the tail side of functional dependencies. *Population* depends on *Country* but also depends on *July 2001*. Knowledge from the data frame library recognizes that the values in the *Religion* columns are *Percent* values. The religions, which could either be object sets or values, are values since there are many (our current threshold is five). Given that religions are values, we therefore have a ternary relationship among *Country*, *Religion*, and *Percent*. Based on constraint mining, we can determine that *Country* and *Religion* together functionally determine *Percent*. Creation of the remaining five mini-ontologies is similar.

15.4.3.3 Mapping Discovery and Ontology Merge

Our approach to discovering inter-ontology mappings is multi-faceted [EJX01, EJX02], which means that we use all evidence at our disposal to determine how to match concepts. These facets include label matching [EJX01], value similarity [EJX01], expected values via matching values with data frames [EJX01, EXD04], constraints [BE03], and structure [ETL02, EJX02]. In using this evidence we look not only for direct matches as is common in most schema matching techniques [BMPQ04, FNS07, RB01] but also for indirect matches [XE06]. Thus, for example, we are able to split or join columns to match the single *Geographic Coordinates* column in Figure 15.11 with the pair of columns, *Lat* and *Lon*, in Figure 15.13, and we are able to divide the values in the *Place* column in Figure 15.13 into several different object sets.

Once we have discovered mappings between mini-ontologies or between a mini-ontology and the ontology we are building, we can begin the merge process. Sometimes the match is such that we can directly fuse two ontologies by simply keeping all the nodes and edges of both and merging nodes and edges that directly correspond. Often, however, merging induces conflicts that must be resolved. We resolve conflicts synergistically based on Issue/Default/Suggestion (IDS) statements [BE03, Lia08] When a conflict arises, the system brings the issue to the attention of a knowledge worker. It provides a default resolution—the one it will take if the user does not intervene, and it makes some suggestions about alternate possibilities. In the tool we have created [Lia08], a user can specify mappings that the automated matching algorithms my miss, can remove mappings that the matching algorithms may have incorrectly suggested, can run the merge automatically (allowing the system to take all the default resolutions for any conflict), can run merge interactively (resolving each IDS statement manually), and can manually adjust the results after merge.

Given a collection of mini-ontologies, such as those in Figure 15.15, we look initially for mini-ontologies that exhibit as large of an overlap as possible (as measured by the number of inter-ontology mappings); thereafter we select mini-ontologies with the largest overlap with our growing ontology. In our example we begin by merging the mini-ontologies in Figures 15(a) and 15(b).

1st Merge *Country* matches *Country* and *Population* matches *Population*. Both *July 2001* and *July 2003* are date components associated with *Population*, and we merge them as *Date*.

2nd Merge Building on the 1st Merge, we add the mini-ontology in Figure 15(d) and obtain the emerging ontology in Figure 15.16. Here, we encounter IDS statements that help us reconcile the lexical/non-lexical *Country* object sets so that *Country* becomes non-lexical with an associated name and also that let *Population* be an inherited property and thus omit it from the *Country* specialization.

3rd Merge Continuing, we merge the mini-ontology in Figure 15(f) with the growing ontology in Figure 15.16. Here, the data in the object sets *Geopolitical Entity* and *Where Spoken* largely overlap, but it is not 100% clear whether one set should be a subset of the other, whether they are overlapping siblings in an is-a hierarchy, or whether they should be the same set. An IDS statement is therefore appropriate, and we assume the issue is resolved by declaring that the sets are the same and should be called *Geopolitical Entity*. This merge thus adds *Region* as a specialization of *Geopolitical Entity* and adds *Language* and *Speakers* connected to *Geopolitical Entity* in the same way they are connected to *Where Spoken (Major)* in Figure 15(f).

4th Merge Continuing, we next add the mini-ontology in Figure 15(c). Here, the constraints on the *Location Description* in Figure 15(c) declare that the relationship is mandatory for both *Country* and *Location Description* and functional from *Country* to *Location Description*. Because of the lack of location descriptions for most countries in our growing collection, however, we have enough evidence to override the mandatory declaration and make the relationship for *Country* optional.

5th Merge Continuing, we next add the mini-ontology in Figure 15(e) and obtain the growing ontology in Figure 15.17. Here, with the help of IDS statements, we must recognize that *Geopolitical Entity* is a specialization of *Place*. Other adjustments come readily, including inheriting *Name* from *Place* and making the existence of *USGS Quad* optional for *Place* based on lack of map locations for most places.

As we transform tables to mini-ontologies and merge them, we also retain the data. The end result is a populated ontology and thus a KB that represents the domain described by the given collection of tables.

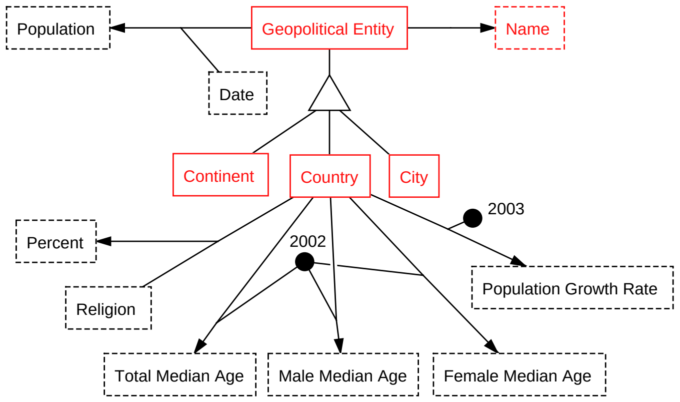


Fig. 15.16 Growing Ontology after Merging the Mini-Ontologies in Figures 15(a), 15(b), and 15(d). (The object sets with lighter-shaded, "red," borders are those most recently added.)

15.4.4 Construction via Form Filling

Although the KB construction methods discussed in Subsections 15.4.1–15.4.3 are largely automatic, they have the disadvantage that users have little or no control over the ontological structure created and the data that populates the ontological structure. Users could take the final generated result and edit it by hand—a reasonable possibility if the desired ontological structure and the data are almost the same as the generated structure and data. In this subsection, we discuss an alternative that gives users a way to create a custom-designed ontological structure for a KB and to populate it with values harvested from a diverse collection of web pages. This method works particularly well when the information to be collected for the KB comes from machine-generated collections of semi-structured web pages such as those commonly found in most hidden-web/deep-web sites.

FOCIH (Forms-based Ontology Creation and Information Harvesting) [TEL09] is a tool that lets users specify ontologies without having to know any conceptual-modeling language or any ontology language. We observe that forms are a natural way for humans to collect information. As an everyday activity, people create forms and ask others to fill in the blanks. FOCIH lets users create their own forms to describe information they wish to harvest. Once defined, users can fill in forms from web pages by copy and paste. From the form specification and user cut-and-paste actions, FOCIH generates an ontology, extracts data, and annotates the web page with respect to the ontology. Further, if the web page is machine-generated and has sibling pages, FOCIH is able to harvest the specified information from all the sibling pages, often without further user intervention.

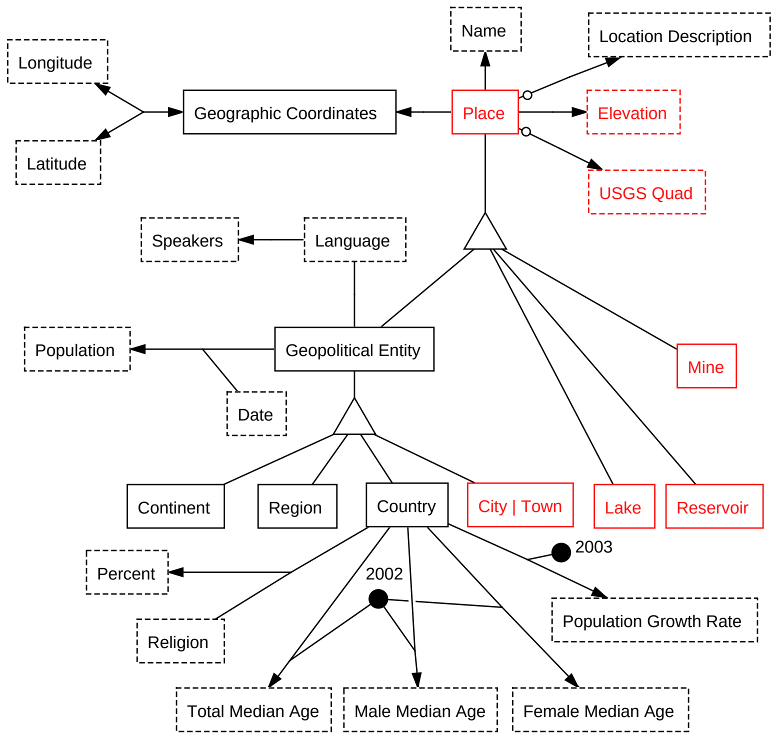
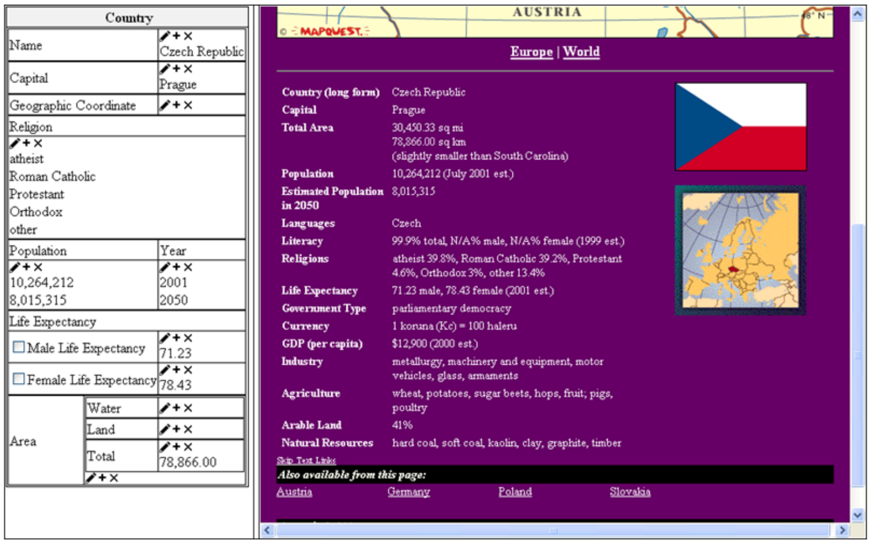


Fig. 15.17 Growing Ontology after Merging all Mini-Ontologies. (The object sets with lighter-shaded, "red," borders are those most recently added.)

FOCIH's form-creation mode provides users with an intuitive method for defining different kinds of form features. FOCIH has five types of form fields: *single-label/single-value*, *single-label/multiple-value*, *multiple-label/multiple-value*, *mutually-exclusive choice*, and *non-exclusive choice*. Users create standard forms by stringing these form elements together in any order and nesting them within one another to any depth. The form in the left panel of Figure 15.18 shows an example. The form is for collecting country information. It starts with three single-label/single-value fields for *Name*, *Capital*, and *Geographic Coordinates*, followed by a single-label/multiple-value field for *Religion*, and a multiple-label/multiple-value field for *Population-Year* estimates. The *Life Expectancy* field is a non-exclusive choice field for either *Male* or *Female Life Expectancy* or both. The final field shows the nesting of three form fields for *Water*, *Land*, and *Total* under a single-label/single-value field for *Area*.

FOCIH's form-fill-in mode lets users browse to a web page they wish to annotate and copy and paste values into form fields. A user highlights values in the page and then clicks on the form field to fill in a value. Figure 15.18 shows a web page for the Czech Republic in the right panel. Copied values



Country

Name		✎+× Czech Republic
Capital		✎+× Prague
Geographic Coordinate		✎+×
Religion		
✎+× atheist, Roman Catholic, Protestant, Orthodox, other		
Population		Year
✎+× 10,264,212, 8,015,315		✎+× 2001, 2050
Life Expectancy		
☐ Male Life Expectancy		✎+× 71.23
☐ Female Life Expectancy		✎+× 78.43
Area	Water	✎+×
	Land	✎+×
	Total	✎+× 78,866.00
	✎+×	

Europe | World

Country (long form)	Czech Republic
Capital	Prague
Total Area	30,450.33 sq mi 78,866.00 sq km (slightly smaller than South Carolina)
Population	10,264,212 (July 2001 est.)
Estimated Population in 2050	8,015,315
Languages	Czech
Literacy	99.9% total, N/A% male, N/A% female (1999 est.)
Religions	atheist 39.8%, Roman Catholic 39.2%, Protestant 4.6%, Orthodox 3%, other 13.4%
Life Expectancy	71.23 male, 78.43 female (2001 est.)
Government Type	parliamentary democracy
Currency	1 koruna (Kc) = 100 haleru
GDP (per capita)	$12,900 (2000 est.)
Industry	metallurgy, machinery and equipment, motor vehicles, glass, armaments
Agriculture	wheat, potatoes, sugar beets, hops, fruit; pigs, poultry
Arable Land	41%
Natural Resources	hard coal, soft coal, kaolin, clay, graphite, timber

Skip Text Links

Also available from this page:

Austria Germany Poland Slovakia

Fig. 15.18 Filled-in FOCIH Form.

from the web page appear in the form in the left panel. The pencil icon lets a user drop a highlighted value into a form field, and the x icon lets a user remove a value. The plus icon lets a user concatenate a second part of the value to a partial value already in the form field. Thus, for example, if the latitude and longitude values are disjoint, perhaps labeled *Latitude:* and *Longitude:* and appearing on separate lines in a web page, a user can concatenate the two as a single value in the *Geographic Coordinates* field.

From the filled-in form, FOCIH can generate a conceptual model and populate it with values. Note that filled-in nested forms are identical in structure to the nested tables discussed in Subsection 15.4.2. Thus, the generated ontologies are similar, and are like the OSM-O model instance in Figure 15.7. In addition to generating a conceptual model and populating it, FOCIH also records the following information: (1) paths to leaf nodes in the DOM tree of an HTML page containing each value and, for concatenated values, each value component; (2) for each value the most specific instance recognizer from the data-frame library (e.g., string, number, percentage, year, geographic coordinate); and (3) enough left, right, and delimiter context within each leaf node to identify the value or values within the DOM-tree node. This enables FOCIH to harvest the same information from other machine-generated sibling pages from the same web site.

The result of running FOCIH over a collection of sibling pages is a custom-built KB containing the information in the collection. Further, FOCIH can harvest information from other sibling-page collections with respect to the same custom-built ontology, which can further augment the KB.

15.5 WoK Usage

The construction of extraction ontologies leads to "understanding" within a WoK. This "understanding" leads to the ability to answer a free-form query because, as we explain in this section, a WoK system can identify an extraction ontology that applies to a query and match the query to the ontology. Hence, a WoK system can reformulate the free-form query as a formal query, so that it can be executed over a KB. In addition, "understanding" leads to establishing a context of discourse, allowing the system to expose its conceptualization of the subject and thus allowing users to more effectively communicate their information needs to the system. In both cases results returned for a query include not only answers to queries but also answer justification. Users can obtain a reasoning chain justifying each answer as well as provenance links identifying each ground fact supporting the answer.

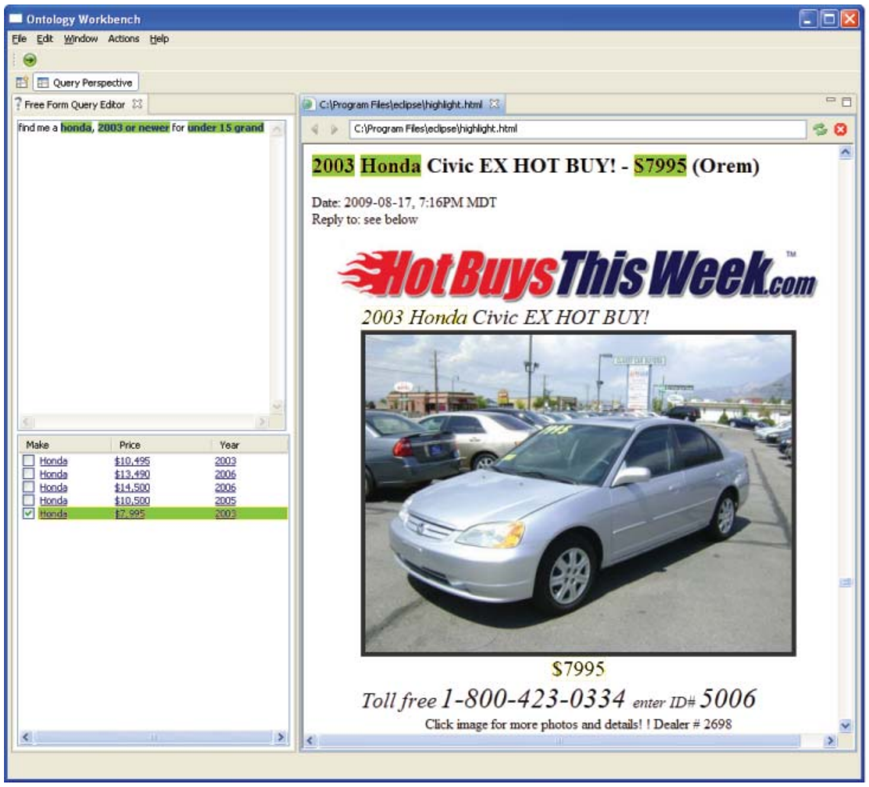
Definition 11. Let S be a source conceptualization and let T be a target conceptualization formalized as an OSM-EO model instance. We say that T *understands* S if there exists an S-to-T transformation that maps each one-place predicate of S to an object set of T, each n-place predicate of S to an n-place relationship set of T ($n \geq 2$), each fact of S to a fact of T with respect to the predicate mappings, and each operator of S to an operator in a data frame of T, such that the constraints of T all hold over the transformed predicates and facts. $\square$

Observe that although Definition 11 states how T is formalized, it does not state how S is formalized. Thus, the predicates and operators of S may or may not be directly specified. This is the hard part of "understanding"—to recognize the applicable predicates and operators. But this is exactly what extraction ontologies are meant to do. If an OSM-EO model instance is linguistically well grounded, then it can "understand" so long as what is stated in S is within the context of T—that is if there is an object set or relationship set in T for every predicate in S and if there is an operator in a data frame of T for every operator in S.

Applications of understanding include free-form query processing, grounded reasoning chains, and KB building for research studies. We explain and illustrate each in turn. In doing so, we also illustrate our WoK prototype system, which we are building as a way to experiment with our vision of a WoK [ELL$^+$08].

15.5.1 Free-Form Query Processing

Figure 15.19 illustrates free-form query processing within our WoK prototype. To "understand" a user query, our WoK prototype first determines which extraction ontology applies to the query by seeing which one recognizes the



find me a honda, 2003 or newer for under 15 grand

	Make	Price	Year
☐	Honda	$10,495	2003
☐	Honda	$13,490	2006
☐	Honda	$14,500	2006
☐	Honda	$10,500	2005
✓	Honda	$7,995	2003

C:\Program Files\eclipse\highlight.html

2003 Honda Civic EX HOT BUY! - $7995 (Orem)

Date: 2009-08-17, 7:16PM MDT
Reply to: see below

2003 Honda Civic EX HOT BUY!

$7995

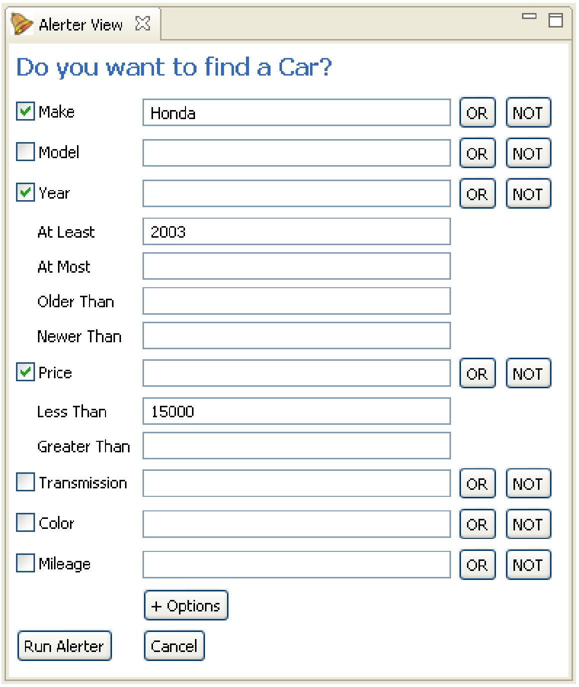
Toll free 1-800-423-0334 enter ID# 5006

Click image for more photos and details! ! Dealer # 2698

Fig. 15.19 Screenshot of WoK Prototype Showing Free-Form Query Processing.

most instances, predicates, and operators in the query request. For the query in Figure 15.19, we assume that the WoK prototype chooses the *Car* extraction ontology illustrated in Figures 15.2 and 15.3. The WoK prototype then applies the S-to-T transformation highlighting what it "understands" ("Find me a honda, 2003 or newer for under 15 grand"). Figure 15.20 shows the result of this transformation—each predicate and each operation is mapped correctly and the constraints of the OSM-EO model instance all hold. Given this "understanding," it is straightforward to generate a SPARQL query. Before executing the query, our WoK prototype augments it so that it also obtains the stored annotation links. Then, when our WoK prototype displays the results of the query (e.g., in the lower-left box in Figure 15.19), it makes returned values clickable. Clicking on a value, causes our WoK prototype to find the page from which the value was extracted, highlight it, and display the page appropriately scrolled to the location that includes the value. The right panel of Figure 15.19 shows several highlighted values, which happens when a user checks one or more check-boxes before clicking. □

The form in Figure 15.20 is for an alerter system which we have implemented for craigslist.org. We use the form in two ways: (1) for comprehensive


Alerter View

Do you want to find a Car?

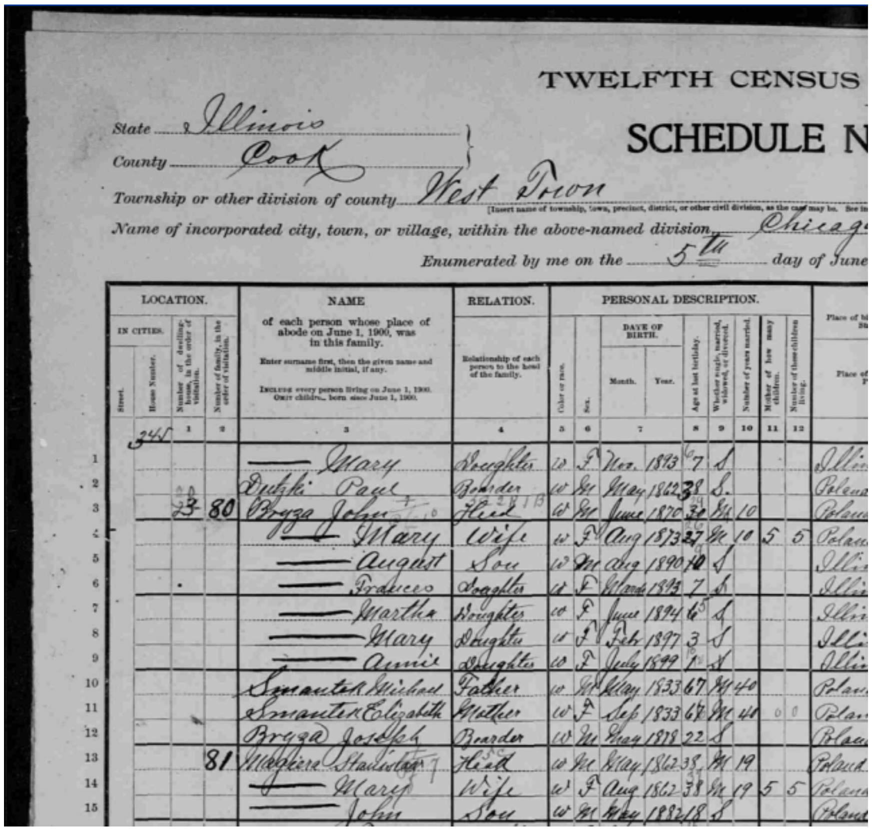
✓ Make	Honda	OR	NOT
☐ Model		OR	NOT
✓ Year		OR	NOT
At Least	2003		
At Most			
Older Than			
Newer Than			
✓ Price		OR	NOT
Less Than	15000		
Greater Than			
☐ Transmission		OR	NOT
☐ Color		OR	NOT
☐ Mileage		OR	NOT

Fig. 15.20 Generated Form Showing the System's "Understanding"—Its "Understood" Instances within Its "Understood" Ontological Context.

feedback to indicate its "understanding" of the query and (2) for giving users advanced options for query specification. As feedback, it lets users know the context in which the system "understands" the query being asked (i.e., the system displays the name of the extraction ontology and its details as form elements), and (2) it lets users know exactly what has been "understood" (i.e., the system displays constant values in fields for object sets or for operations applicable to object sets). With respect to advanced options, it lets users know what else can be asked in the context of the query. A user then has the opportunity to adjust the query or add additional constraints. For example, besides Hondas, a user may wish to also know if Toyotas are for sale but only if they if they are not Camrys. Clicking on *OR* for *Make* and adding *Toyota* and then clicking on *NOT* for *Model* and adding *Camry* makes this possible. The plus icons show that more operators are available; clicking on the plus displays them. For example, a user may wish to limit the search to cars whose odometer reading is less than 100K miles; clicking on the "+ Options" button shows the Boolean operators for *Mileage* and lets a user enter this limitation.

15.5.2 Grounded Reasoning Chains

To illustrate grounded reasoning chains, we give an example from family-history research. Many millions of hand-written records such as those in the census record in Figure 15.21 have been transcribed by human indexers [Fam]. Using extraction ontologies, we can extract from the transcription to associate names, dates, places, and other information with a genealogical ontology. Bounding boxes for names and other information in the image are also available, so we know where in the image information appears.



TWELFTH CENSUS

SCHEDULE N

State Illinois

County Cook

Township or other division of county West Town

[Insert name of township, town, precinct, district, or other civil division, as the case may be.

Name of incorporated city, town, or village, within the above-named division Chicag

Enumerated by me on the 5th day of June

LOCATION. IN CITIES. Street.	House Number.	Number of dwelling-house, in the order of visitation.	Number of family, in the order of visitation.	NAME of each person whose place of abode on June 1, 1900, was in this family.	RELATION. Relationship of each person to the head of the family.	Color or race.	Sex.	DATE OF BIRTH. Month.	Year.	Age at last birthday.	Whether single, married, widowed, or divorced.	Number of years married.	Mother of how many children.	Number of these children living.	Place of
		1	2	3	4	5	6	7		8	9	10	11	12	
1	345			—— Mary	Daughter	w	F	Nov.	1893	7	S				Illi
2				Dutzki Paul	Boarder	w	M	May	1862	38	S.				Polan
3		23	80	Bryza John	Head	w	M	June	1870	30	M	10			Polan
4				—— Mary	Wife	w	F	Aug	1873	27	M	10	5	5	Polan
5				—— August	Son	w	M	Aug	1890	10	S				Illi
6				—— Frances	Daughter	w	F	March	1893	7	S				Illi
7				—— Martha	Daughter	w	F	June	1894	6	S				Illi
8				—— Mary	Daughter	w	F	Feb	1897	3	S				Illi
9				—— Annie	Daughter	w	F	July	1899	1	S				Illi
10				Smantek Michael	Father	w	M	May	1833	67	M	40			Polan
11				Smantek Elizabeth	Mother	w	F	Sep	1833	67	M	40	0	0	Polan
12				Bryza Joseph	Boarder	w	M	May	1878	22	S				Polan
13			81	Magiera Stanislaw	Head	w	M	May	1862	38	M	19			Poland
14				—— Mary	Wife	w	F	Aug	1862	38	M	19	5	5	Polan
15				—— John	Son	w	M	May	1882	18	S				Polan

Fig. 15.21 Census Record.

It is not hard to see that, among others, the following rules hold and are useful for establishing family relationships implied by the information in the census record in Figure 15.21:

```
Person(x)isHusbandOfPerson(y) :- Person(x), Person(y),
        Person(x)hasGender('Male'),
        Person(x)hasRelationToHead('Head'),
        Person(y)hasRelationToHead('Wife'),
        Person(x)isInSameFamilyAsPerson(y).
Person(x)isInSameFamilyAsPerson(y) :-
        Person(x)hasFamilyNumber(z)inCensusRecord(w),
        Person(y)hasFamilyNumber(z)inCensusRecord(w).
Person(x)named(y)isHusbandOfPerson(z)named(w) :-
        Person(x)isHusbandOfPerson(z), Person(x)hasName(y),
        Person(z)hasName(w).
```

The first rule states that a person x is the husband of person y if x is a male head of the family, y is the wife, and x and y are in the same family. The second rule assures that they are in the same family by checking to see that their family number is the same, and the last rule associates the husband and wife with their names.

When we associate a rule with an ontology, we must ensure that it is grounded in the base predicates of the ontology. The set of rules forms a graph over a "head predicate depends-on body predicate" relation, and this graph must lead to predicates declared in the ontology as object-set predicates, relationship-set predicates or Boolean operations declared for the ontology. Recursive rules such as the following rules to compute ancestors are possible, but must also be grounded.

```
  Person(x)isAncestorOfPerson(y) :-
          Person(x)isParentOfPerson(y).
  Person(x)isAncestorOfPerson(y) :-
          Person(x)isParentOfPerson(z),
          Person(z)isAncestorOfPerson(y).
```

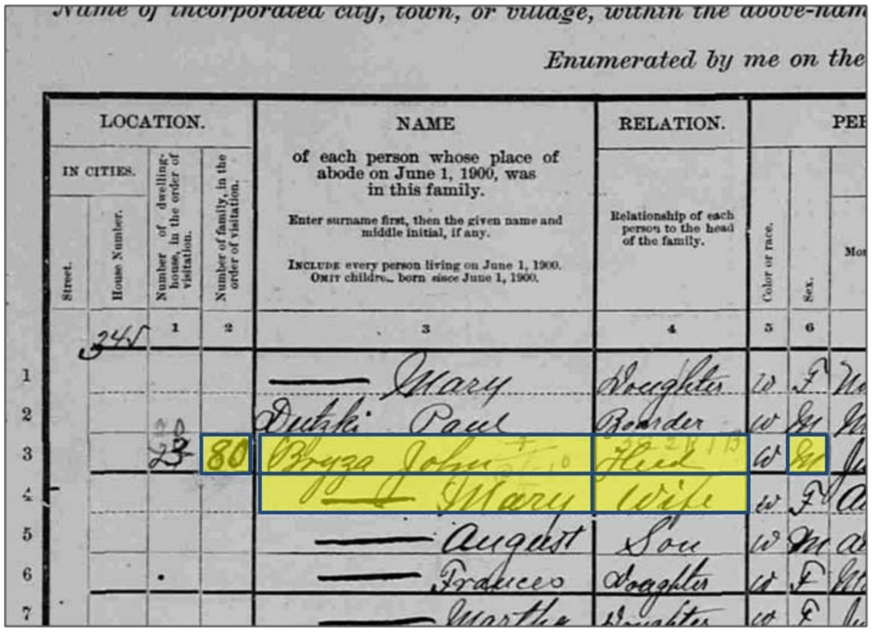
Here, the predicate $Person(x)isParentOfPerson(y)$ must be a rule head that eventually resolves down to ground predicates such as $Person(x)has$-$RelationToHead('Son')$ or $Person(x)hasRelationToHead('Daughter')$.

We linguistically ground a rule r by declaring a data frame for r in the same way we declare a data frame for an object set (if the head of r is a one-place predicate) or for a relationship set (if the head of r is an n-place predicates, $n \geq 2$). Thus, for example, $\{Person\}\backslash s{*}is\backslash .{*}husband\backslash s{*}of\backslash s{*}\{Person\}$ may be one of the regular expressions for the rule head *Person(x)isHusbandOf-Person(y)*.

Now when we ask the query, "Who is the husband of Mary Bryza?", we can match the query to our genealogy ontology and specifically to the rule and thus also the chain of rules needed to answer the query. The returned result would yield "John Bryza" and perhaps others if other Mary Bryzas are known within the KB. Clicking on "John Bryza" yields both the reasoning chain in Figure 15.22 and the highlighted census recored in Figure 15.23. The reasoning chain is simply a list of rules invoked with instance data filled in for the variables and reformatted to be more readable. The highlighted census record shows the source of all the extracted ground fact values used to yield the answer.

Person(p1) named('John Bryza') is husband of Person(p2) named('Mary Bryza')
because:
  Person(p1) is husband of Person(p2) and Person(p1) has Name('John Bryza')
    and Person(p2) has Name('Mary Bryza');
and Person(p1) is husband of Person(p2)
because:
  Person(p1) has gender('Male') and Person(p1) has relation to Head('Head'),
    and Person(p2) has relation to Head('Wife')
    and Person(p1) is in same family as Person(p2).
and Person(p1) is in same family as Person(p2)
because:
  Person(p1) has family number(80) in Census Record(r1)
    and Person(p2) has family number(80) in Census Record(r1).

Fig. 15.22 Reasoning Chain for Query.



Name of incorporated city, town, or village, within the above-nam

Enumerated by me on the

	LOCATION.				NAME	RELATION.		
	IN CITIES.				of each person whose place of abode on June 1, 1900, was in this family. Enter surname first, then the given name and middle initial, if any. INCLUDE every person living on June 1, 1900. OMIT children born since June 1, 1900.	Relationship of each person to the head of the family.		
	Street.	House Number.	Number of dwelling-house, in the order of visitation.	Number of family, in the order of visitation.			Color or race.	Sex.
			1	2	3	4	5	6
		245						
1					— Mary	Daughter	W	F
2					Dutki Paul	Boarder	W	M
3			23	80	Bryza John	Head	W	M
4					— Mary	Wife	W	F
5					— August	Son	W	M
6					— Frances	Daughter	W	F
7					— Martha	Daughter	W	F

Fig. 15.23 Census Record with Ground Facts Highlighted.

15.5.3 Knowledge Bundles for Research Studies

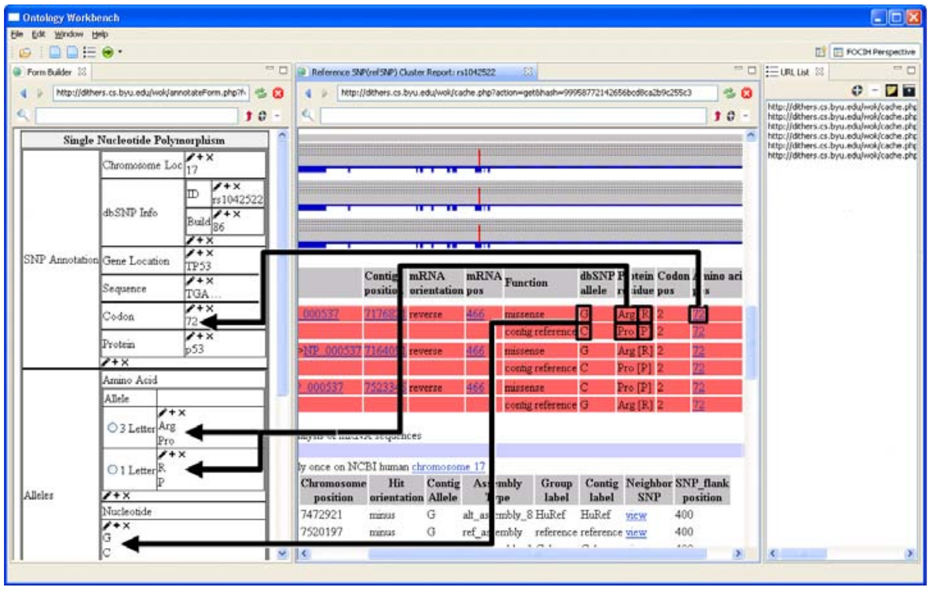
In addition to "understanding" queries, it should be clear that "understanding" is also about fact finding. The fundamental intent of linguistically grounding extraction ontologies is to allow them to recognize facts in structured, semi-structured, and unstructured text. As an example, we give a plausible scenario, based on the WoK components we have presented, for gathering facts for a bio-research study and storing them as a KB for further analysis [ELL$^+$09]. Gathering tasks for these research studies often take trained bio-researchers several man-months of work. So, any significant speed-up extraction ontologies can provide would be of great benefit in bio-medical research.

Suppose a bio-researcher B wishes to study the association of TP53 polymorphism and lung-cancer. To do this study, B wants information from the NCBI dbSNP repository[3] about SNPs (chromosome location, SNP ID and build, gene location, codon, and protein), about alleles (amino acids and nucleotides), and about the nomenclature for amino-acid levels and nucleotide levels. B also needs data about human subjects with lung cancer and needs to relate the SNP information to human-subject information.

To gather information from dbSNP, B uses FOCIH to construct the form in the left panel in Figure 15.24. Form construction consists of selecting types of form fields and organizing and nesting form fields so that they are a conceptualization of the information B wishes to harvest for the research study. B next finds a first SNP page in dbSNP from which to begin harvesting information. (The created form and located page need not have any special correspondence—no schema correspondence, no name correspondence, and no special structure requirements—but, of course, the page should have data of interest for the research study and thus for the created form.) B then fills in the form by cut-and-paste actions, copying data from the page in the center panel in Figure 15.24 to the form in the left panel.

To harvest similar information from the numerous other dbSNP pages, B gives a list of URLs, as the right panel in Figure 15.24 illustrates (although there would likely be hundreds rather than just the six in Figure 15.24). The FOCIH system automatically harvests the desired information from the dbSNP pages referenced in the URL list. Since one of the challenges bio-researchers face is searching through the pages to determine which ones contain the desired information, FOCIH should provide a filtering mechanism. By adding constraints to form fields, bio-researchers can cause the FOCIH harvester to gather information only from pages that satisfy the constraints. B, for example, might only want coding SNP data with a significant heterogeneity (i.e., minor allele frequency $> 1\%$).

[3] The Single Nucleotide Polymorphism database (dbSNP) is a public-domain archive for a broad collection of simple genetic polymorphisms hosted by National Center for Biotechnology Information (NCBI) at www.ncbi.nlm.nih.gov/projects/SNP/.


Ontology Workbench
File Edit Window Help
Form Builder
http://dithers.cs.byu.edu/wok/annotateForm.php?f
Reference SNP(refSNP) Cluster Report: rs1042522
http://dithers.cs.byu.edu/wok/cache.php?action=get&hash=99958772142656bcd8ca2b9c255c3
URL List
FOCIH Perspective

Single Nucleotide Polymorphism			
SNP Annotation	Chromosome Loc		17
	dbSNP Info	ID	rs1042522
		Build	86
	Gene Location		TP53
	Sequence		TGA…
	Codon		72
	Protein		p53
Alleles	Amino Acid		
	Allele		
		3 Letter	Arg Pro
		1 Letter	R P
	Nucleotide		G C

	Contig positio	mRNA orientation	mRNA pos	Function	dbSNP allele	P tein r idue		Codon pos	nino aci s
000537	717682	reverse	466	missense	G	Arg	[R]	2	72
				contig reference	C	Pro	[P]	2	72
>NP_000537	716405	reverse	466	missense	G	Arg	[R]	2	72
				contig reference	C	Pro	[P]	2	72
000537	752334	reverse	466	missense	C	Pro	[P]	2	72
				contig reference	G	Arg	[R]	2	72

ly once on NCBI human chromosome 17

Chromosome position	Hit orientation	Contig Allele	Assembly Type	Group label	Contig label	Neighbor SNP	SNP_flank position
7472921	minus	G	alt_assembly_8	HuRef	HuRef	view	400
7520197	minus	G	ref_assembly	reference	reference	view	400

http://dithers.cs.byu.edu/wok/cache.php
http://dithers.cs.byu.edu/wok/cache.php
http://dithers.cs.byu.edu/wok/cache.php
http://dithers.cs.byu.edu/wok/cache.php
http://dithers.cs.byu.edu/wok/cache.php
http://dithers.cs.byu.edu/wok/cache.php

Fig. 15.24 Form Filled in with Information from an SNP Page.

For the research scenario, B may also wish to harvest information from other sites such as GeneCard. B can use FOCIH with the same form to harvest from as many sites as desired. Interestingly, however, once FOCIH harvests from one site, it can use the knowledge it has already gathered to do some of the initial cut-and-paste for B. In addition to just being a structured knowledge repository, the KB being produced also becomes an extraction ontology capable of recognizing data items it has already seen. It can also recognize data items it has not seen but are like the data it has seen—e.g., numeric values or DNA snippets.

Using KBs as extraction ontologies also lets bio-researchers search the literature. Suppose B wishes to find papers related to the information harvested from the dbSNP pages. B can point the extraction ontology to a repository of papers to search and cull out those that are relevant to the study. Using the KB as an extraction ontology provides a sophisticated query of the type used in information retrieval resulting in high-precision document filtering. For example, the extraction ontology recognizes the highlighted words and phrases in the portion of the paper in Figure 15.25. With the high density of not only keywords but also data values and relationships all aligned with the ontological KB, the system can designate this paper as being relevant for B's study.

For collecting human-subject information, B may decide to obtain information from Indivo, a database containing personally controlled health records. Based on reverse-engineering techniques, the system can automat-

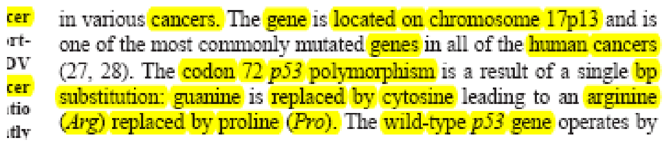
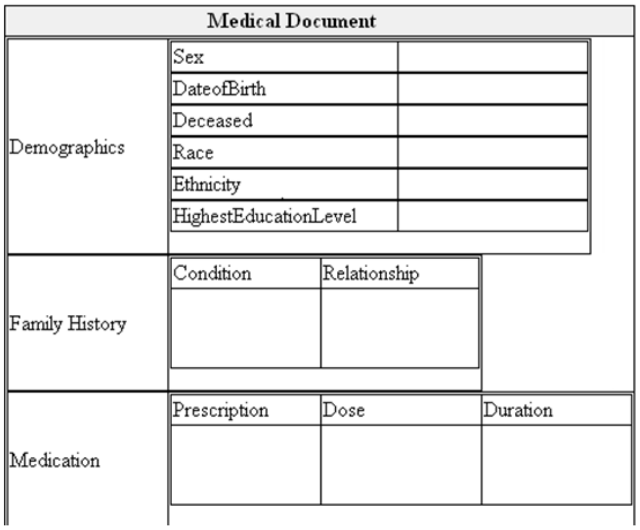

in various cancers. The gene is located on chromosome 17p13 and is one of the most commonly mutated genes in all of the human cancers (27, 28). The codon 72 *p53* polymorphism is a result of a single bp substitution: guanine is replaced by cytosine leading to an arginine (*Arg*) replaced by proline (*Pro*). The wild-type *p53* gene operates by

Fig. 15.25 Paper Retrieved from PMID Using an Extraction Ontology.


Medical Document

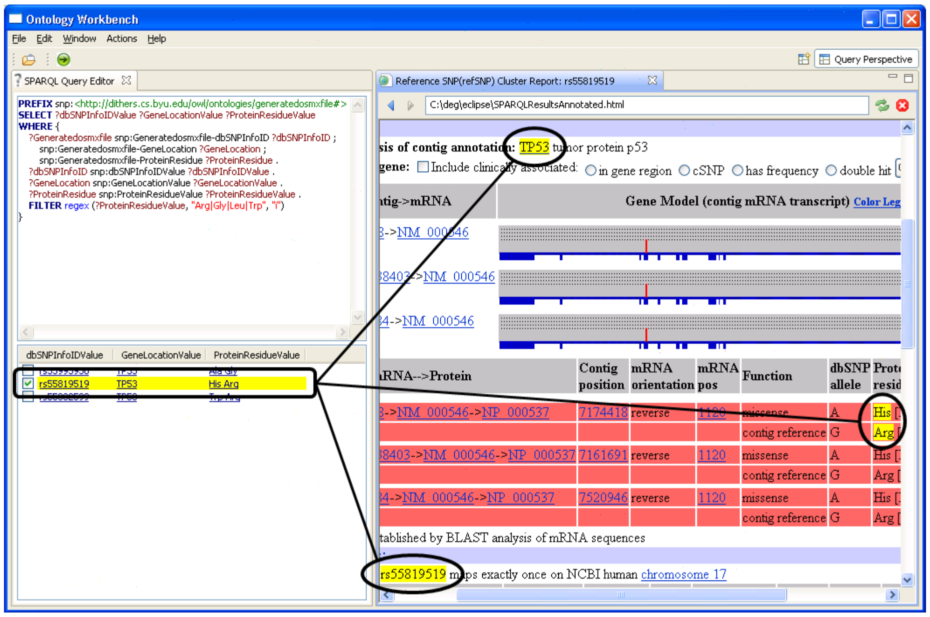
Demographics	Sex	
	DateofBirth	
	Deceased	
	Race	
	Ethnicity	
	HighestEducationLevel	

Family History	Condition	Relationship	
Medication	Prescription	Dose	Duration

Fig. 15.26 Some Human Subject Information Reverse-Engineered from Indivo.

ically reverse-engineer the Indivo database to a KB, and present B with a form representing the schema of the database. Figure 15.26 shows an example. B can then modify the form, deleting fields not of interest and rearranging fields to suit the needs of the study. Further, B can add constraints to the fields so that the the systems only gathers data of interest.

With all information harvested and organized into a KB, B can now issue queries and reason about the data to do some interesting analysis. Figure 15.27 shows a sample SPARQL query over the data harvested from the pages referenced by the six URLs listed in Figure 15.24. The query finds three SNPs that satisfy the query's criteria and for each, returns the dbSNP ID, the gene location, and the protein residue. As Figure 15.27 shows, B wishes to see the source of the query result <*rs55819519*, *TP53*, *His Arg*>.


Ontology Workbench
File Edit Window Actions Help
Query Perspective
SPARQL Query Editor
Reference SNP(refSNP) Cluster Report: rs55819519
C:\deg\eclipse\SPARQLResultsAnnotated.html

```
PREFIX snp:<http://dithers.cs.byu.edu/owl/ontologies/generatedosmxfile#>
SELECT ?dbSNPInfoIDValue ?GeneLocationValue ?ProteinResidueValue
WHERE {
  ?Generatedosmxfile snp:Generatedosmxfile-dbSNPInfoID ?dbSNPInfoID ;
    snp:Generatedosmxfile-GeneLocation ?GeneLocation ;
    snp:Generatedosmxfile-ProteinResidue ?ProteinResidue .
  ?dbSNPInfoID snp:dbSNPInfoIDValue ?dbSNPInfoIDValue .
  ?GeneLocation snp:GeneLocationValue ?GeneLocationValue .
  ?ProteinResidue snp:ProteinResidueValue ?ProteinResidueValue .
  FILTER regex (?ProteinResidueValue, "Arg|Gly|Leu|Trp", "i")
}
```

	dbSNPInfoIDValue	GeneLocationValue	ProteinResidueValue
☐	[illegible]	TP53	Ala Gly
✓	rs55819519	TP53	His Arg
☐	[illegible]	TP53	Trp Arg

sis of contig annotation: TP53 tumor protein p53

gene: ☐ Include clinically associated: ○ in gene region ○ cSNP ○ has frequency ○ double hit

ntig->mRNA	Gene Model (contig mRNA transcript) Color Leg
3->NM_000546	
38403->NM_000546	
34->NM_000546	

mRNA-->Protein	Contig position	mRNA orientation	mRNA pos	Function	dbSNP allele	Prote resid
3->NM_000546->NP_000537	7174418	reverse	1120	missense	A	His [
				contig reference	G	Arg [
38403->NM_000546->NP_000537	7161691	reverse	1120	missense	A	His [
				contig reference	G	Arg [
34->NM_000546->NP_000537	7520946	reverse	1120	missense	A	His [
				contig reference	G	Arg [

tablished by BLAST analysis of mRNA sequences

rs55819519 maps exactly once on NCBI human chromosome 17

Fig. 15.27 Screenshot of our Web of Knowledge Prototype.

15.6 Conclusion

We have described a web of knowledge (WoK) as a collection of interconnected KBs superimposed over a web of documents. Our WoK vision has conceptual modeling at its foundation. As described, a WoK consists of KBs, which are conceptual-model instances augmented with facilities that provide (1) for both extensional and intensional facts, (2) for linking between KBs yielding a web of data, and (3) for authentication by linking to source documents and explicating reasoning chains.

We have provided a formal foundation for WoK components—ontologies (OSM-O) in terms of decidable first-order logic and extraction ontologies (OSM-EO) linguistically grounded via data-frame recognizers. In addition, we have formalized a WoK as a collection of interconnected knowledge bundles (KBs) consisting of OSM-EO model instances with valid interpretations super-imposed over source documents.

Further, we have addressed concerns about WoK construction. Transformations map source conceptualizations to target conceptualizations. Information- and constraint-preserving transformations guarantee that target conceptualizations completely and accurately capture source conceptualizations. We have explained how reverse engineering of some documents can yield source conceptualizations guaranteed to preserve information and constraints. We conclude, however, that many source conceptualizations (ranging from semi-structured sources such as ordinary human-readable tables and forms to unstructured sources such as free-running text) likely require best-effort automation methods and may involve some user supervision. We have given, as examples, a way to transform a collection of ordinary tables with overlapping information from some domain into a KB and a way to construct KBs via form-creation and form-filling.

Finally, we have addressed concerns about WoK usage. When transformations exist that map source predicates and operations to an established ontology, the ontology is said to have "understood" the information in the source. "Understanding" applied to free-form queries allows untrained users to query the envisioned WoK. Users receive direct answers to queries, rather than pages that may contain answers. They may, however, ask for justification by clicking on displayed answers, which yields the pages from which the answers were taken and also yields an explanation of any reasoning used to generate inferred answers for the query. As another example of WoK usage, we have illustrated the process of creating a KB for bio-medical research studies.

We have implemented a WoK prototype [ELL$^+$08] including some prototypical extraction ontologies [ECJ$^+$99]. We have also done some work on automated extraction-ontology construction [TE09, LE09, TEL09, TEL$^+$05] and some work on free-form query processing [Vic06, AME07]. We nevertheless still have much work to do, even on fundamental WoK components such as creating a sharable data-frame library, constructing data frames for relationship sets, finding ways to more easily produce instance recognizers, developing processes for reverse-engineering additional genres of semi-structured sources into KBs, investigating bootstrapping as a way to construct extraction ontologies, enhancing query processing, incorporating reasoning, and addressing performance scalability. We also see many opportunities for incorporating the vast amount of work done by others on information extraction, information integration, and record linkage. We cite as relevant examples: KnowItAll [ECD$^+$05], OMNIVORE [Caf09], best-effort information extraction [SDM$^+$08], C-PANKOW [CLS05], Q/A systems [RFRF08], bootstrapping pay-as-you-go data integration [SDH08], large-scale deduplication [ARS09], and OpenDMAP [HLF$^+$08].

These collective efforts will eventually lead to a WoK—a realization of ideas of visionaries from Bush [Bus45] to Berners-Lee [BLHL01] and Weikum [WKRS09]. Conceptual modeling can and should play a foundational role in these efforts.

Acknowledgements: Many students and colleagues have contributed in various ways to the web-of-knowledge vision presented here—as colleagues: Douglas M. Campbell, George Nagy, Dennis Ng, Dallan Quass, Yuri Tijerino; as PhD students: Reema Al-Kamha, Muhammed Al-Muhammed, Yihong Ding, Aaron Stewart, Cui Tao, Li Xu; as MS students: Tim Chartrand, Wendy Chen, Xueqi Helen Chen, Kimball Hewett, David Jackman, Stephen Jiang, Zonghui Lian, Stephen Lynn, Rob Lyon, Lars Olson, Craig Parker, Clint Tustison, Randy Smith, Mark Vickers, Troy Walker, Alan Wessman, Charla Woodbury, Sai Ho Tony Yau, Yuanqiu Joe Zhou, Andrew Zitzelberger; and as undergraduate research assistants: Robert Clawson, Derek McDaniel, Jeff Peters, Robby Watts. We appreciate the support of the National Science Foundation, whose grants have assisted many of these students in their educational endeavors (Grant Numbers: IIS0083127 & IIS0414644).

References

[AH08] D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann Publishers, Burlington, Massachusetts, 2008.

[Aik98] P.H. Aiken. Reverse engineering of data. *IBM Systems Journal*, 37(2):246–269, 1998.

[AK07] R. Al-Kamha. *Conceptual XML for Systems Analysis*. PhD dissertation, Brigham Young University, Department of Computer Science, June 2007.

[AME07] M. Al-Muhammed and D.W. Embley. Ontology-based constraint recognition for free-form service requests. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE'07)*, pages 366–375, Istanbul, Turkey, April 2007.

[AriBC] Aristotle. *Metaphysics*. Oxford University Press, New York, about 350BC. (1993 translation).

[ARS09] A. Arasu, C. Re, and D. Suciu. Large-scale deduplication with constraints using Dedupalog. In *Proceedings of the 25th International Conference on Data Engineering (ICDE 2009)*, pages 952–963, Shanghi, China, March/April 2009.

[BCHS09] P. Buitelaar, P. Cimiano, P. Haase, and M. Sintek. Towards linguistically grounded ontologies. In *Proceedings of the 6th European Semantic Web Conference (ESWC'09)*, pages 111–125, Heraklion, Greece, May/June 2009.

[BE03] J. Biskup and D.W. Embley. Extracting information from heterogeneous information sources using ontologically specified target views. *Information Systems*, 28(3):169–212, 2003.

[BL99] T Berners-Lee. *Weaving the Web*. Harper SanFrancisco, San Francisco, California, 1999.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 36(25):34–43, May 2001.

[BMPQ04] P.A. Bernstein, S. Melnik, M. Petropoulos, and C. Quix. Industrial-strength schema matching. *SIGMOD Record*, 33(4):38–43, December 2004.

[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.

[BN08] J. Bleiholder and F. Naumann. Data fusion. *ACM Computing Surveys*, 41(1):1–41, December 2008.

[Bus45] V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, July 1945.

[Caf09] M.J. Cafarella. Extracting and querying a comprehensive web database. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR'09)*, Monterey, California, 2009.

[CGL09] A. Calì, G. Gottlob, and T. Lukasiewicz. Tractable query answering over ontologies with Datalog$^{\pm}$. In *Proceedings of the DL Home 22nd International*

Workshop on Description Logics (DL 2009), Oxford, United Kingdom, July 2009.

[CGP09] A. Cali, G. Gottlob, and A. Pieris. Tractable query answering over conceptual schemata. In *Proceedings of the 28th International Conference on Conceptual Modeling (ER 2009)*, pages 175–190, Gramado, Brazil, November 2009.

[Cim06] P. Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer Verlag, New York, New York, 2006.

[CLS05] P. Cimiano, G. Ladwig, and S. Staab. Gimme' the context: Context-driven automatic semantic annotation with C-PANKOW. In *Proceedings of the 14th International World Wide Web Conference (WWW205)*, pages 332–341, Chiba, Japan, May 2005.

[dSJT06] A.C. de Silva, A.M. Jorge, and L. Torgo. Design of an end-to-end method to extract information from tables. *International Journal of Document Analysis and Recognition*, 8(2):144–171, 2006.

[ECD+05] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[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.

[EHLN06] D.W. Embley, M. Hurst, D. Lopresti, and G. Nagy. Table-processing paradigms: A research survey. *International Journal of Document Analysis*, 8(2):66–86, 2006.

[EIV07] A.K. Elmagarmid, P.G. Ipeirotis, and V.S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):1–16, January 2007.

[EJX01] D.W. Embley, D. Jackman, and L. Xu. Multifaceted exploitation of metadata for attribute match discovery in information integration. In *Proceedings of the International Workshop on Information Integration on the Web (WIIW'01)*, pages 110–117, Rio de Janeiro, Brazil, April 2001.

[EJX02] D.W. Embley, D. Jackman, and L. Xu. Attribute match discovery in information integration: Exploiting multiple facets of metadata. *Journal of the Brazilian Computing Society*, 8(2):32–43, November 2002.

[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.

[ELL+08] D.W. Embley, S.W. Liddle, D. Lonsdale, G. Nagy, Y. Tijerino, R. Clawson, J. Crabtree, Y. Ding, P. Jha, Z. Lian, S. Lynn, R.K. Padmanabhan, J. Peters, C. Tao, R. Watts, C. Woodbury, and A. Zitzelberger. A conceptual-model-based computational alembic for a web of knowledge. In *Proceedings of the 27th International Conference on Conceptual Modeling (ER08)*, pages 532–533, Barcelona, Spain, October 2008.

[ELL+09] D.W. Embley, S.W. Liddle, D.W. Lonsdale, A. Stewart, and C. Tao. KBB: A knowledge-bundle builder for research studies. In *Proceedings of 2nd International Workshop on Active Conceptual Modeling of Learning (ACM-L 2009)*, Gramado, Brazil, November 2009. (to appear).

[Emb80] D.W. Embley. Programming with data frames for everyday data items. In *Proceedings of the 1980 National Computer Conference*, pages 301–305, Anaheim, California, May 1980.

[ES07] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag, Heidelberg, Germany, 2007.

[ETL02] D.W. Embley, C. Tao, and S.W. Liddle. Automatically extracting ontologically specified data from HTML tables with unknown structure. In *Proceedings of* *the 21st International Conference on Conceptual Modeling (ER2002)*, pages 322–327, Tampere, Finland, October 2002.

[EXD04] D.W. Embley, L. Xu, and Y. Ding. Automatic direct and indirect schema mapping: Experiences and lessons learned. *SIGMOD Record*, 33(4):14–19, December 2004.

[Fam] Family search. http://familysearch.org.

[FM09] J.R. Finkel and C.D. Manning. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore, August 2009.

[FNS07] S.M. Falconer, N.F. Noy, and M.-A. Storey. Ontology mapping—a user survey. In *Proceedings of the Second International Workshop on Ontology Mapping (OM-2007)*, pages 49–60, Bexco, Busan, Korea, November 2007.

[GBH+07] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak. Towards domain-independent information extraction from web tables. In *Proceedings of the Sixteenth International World Wide Web Conference (WWW2007)*, pages 71–80, Banff, Alberta, Canada, May 2007.

[GGTB08] C. Grover, S. Givon, R. Tobin, and J. Ball. Named entity recognition for digitised historical texts. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, May 2008.

[Gru93] T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[Gua98] N. Guarino. Formal ontologies and information systems. In N. Guarino, editor, *Proceedings of the First International Conference on Formal Ontology in Information Systems (FOIS98)*, pages 3–15, Trento, Italy, June 1998.

[HLF+08] L. Hunter, Z. Lu, J. Firby, W.A. Baumgartner Jr., H.L. Johnson, P.V. Ogren, and K.B. Cohen. OpenDMAP: An open source, ontology-driven, concept analysis engine, with applications to capturing knowledge regarding protein transport, protein interactions and cell-type-specific gene expression. *BMC Bioinformatics*, 9(8), 2008.

[HRO06] A. Halevy, A. Rajaraman, and J. Ordille. Data integration: The teenage years. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06)*, pages 9–16, Seoul, Korea, September 2006.

[JN08] P. Jha and G. Nagy. Wang notation tool: Layout independent representation of tables. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR08)*, Tampa, Florida, December 2008.

[JNS+09] R.C. Jandhyala, G. Nagy, S. Seth, W. Silversmith, M. Krishnamoorthy, and R. Padmanabhan. From tessellations to table interpretation. In *Lecture Notes in Artificial Intelligence*, volume 5625, pages 422–437, Berlin, 2009. Springer-Verlag.

[LE09] S. Lynn and D.W. Embley. Semantically conceptualizing and annotating tables. In *Proceedings of the Third Asian Semantic Web Conference*, pages 345–359, Bangkok, Thailand, February 2009.

[Lia08] Z. Lian. A tool to support ontology creation based on incremental mini-ontology merging. Master's thesis, Department of Computer Science, Brigham Young University, Provo, Utah, March 2008.

[PJK+09] R. Padmanabhan, R.C. Jandhyala, M. Krishnamoorthy, G. Nagy, S. Seth, and W. Silversmith. Interactive conversion of large web tables. In *Proceedings of Eighth International Workshop on Graphics Recognition (GREC 2009)*, pages 32–43, La Rochelle, France, July 2009.

[PlaBC] Plato. *Theaetetus*. BiblioBazaar, LLC, Charleston, South Carolina, about 360BC. (translated by Benjamin Jowett).

[PSC+07] A. Pivk, Y. Sure, P. Cimiano, M. Gams, V. Rajkovič, and R. Studer. Transforming arbitrary tables into logical form with TARTAR. *Data & Knowledge Engineering*, 60:567–595, 2007.

[RB01] E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001.

[RFRF08] D. Roussinov, W. Fan, and J. Robles-Flores. Beyond keywords: Automated question answering on the web. *Communications of the ACM*, 51(9):60–65, September 2008.

[RK06] F. Rahman and B. Klein. Special issue on detection and understanding of tables and forms for document processing applications. *International Journal of Document Analysis*, 8(2):65, 2006.

[Ros05] R. Rosati. On the decidability and complexity of integrating ontologies and rules. *Journal of Web Semantics*, 3(1):61–73, 2005.

[Sar08] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[SDH08] A.D. Sarma, X. Dong, and A. Halevy. Bootstrapping pay-as-you-go data integration systems. In *Proceedings of SIGMOD'08*, pages 861–874, Vancouver, British Columbia, Canada, June 2008.

[SDM$^+$08] W. Shen, P. DeRose, R. McCann, A. Doan, and R. Ramakrishnan. Toward best-effort information extraction. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1031–1042, Vancouver, British Columbia, Canada, June 2008.

[SHBL06] N. Shadbolt, W. Hall, and T. Berners-Lee. The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101, May/June 2006.

[Smi03] B. Smith. Ontology. In L. Floridi, editor, *Blackwell Guide to the Philosophy of Computing and Information*, pages 155–166. Oxford: Blackwell, 2003.

[Tao08] C. Tao. *Ontology Generation, Information Harvesting and Semantic Annotation for Machine-Generated Web Pages*. PhD dissertation, Brigham Young University, Department of Computer Science, December 2008.

[TE09] C. Tao and D.W. Embley. Automatic hidden-web table interpretation, conceptualization, and semantic annotation. *Data & Knowledge Engineering*, 68(7):683–703, July 2009.

[TEL$^+$05] Y.A. Tijerino, D.W. Embley, D.W. Lonsdale, Y. Ding, and G. Nagy. Toward ontology generation from tables. *World Wide Web: Internet and Web Information Systems*, 8(3):261–285, September 2005.

[TEL09] C. Tao, D.W. Embley, and S.W. Liddle. FOCIH: Form-based ontology creation and information harvesting. In *Proceedings of the 28th International Conference on Conceptual Modeling (ER2009)*, pages 346–359, Gramado, Brazil, November 2009.

[Vic06] M. Vickers. Ontology-based free-form query processing for the semantic web. Master's thesis, Brigham Young University, Provo, Utah, June 2006.

[W3C] W3C (World Wide Web Consortium) *Semantic Web Activity Page*. http://www.w3.org/2001/sw/.

[Wan96] X. Wang. *Tabular Abstraction, Editing, and Formatting*. PhD thesis, University of Waterloo, 1996.

[WCW$^+$09] J. Wang, C. Chen, C. Wang, J. Pei, J. Bu, Z. Guan, and W.V. Zhang. Can we learn a template-independent wrapper for news article extraction from a single training site? In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1345–1354, Paris, France, June/July 2009.

[WKRS09] G. Weikum, G. Kasneci, M. Ramanath, and F. Suchanek. Database and information-retrieval methods for knowledge discovery. *Communications of the ACM*, 52(4):56–64, April 2009.

[XE06] L. Xu and D.W. Embley. A composite approach to automating direct and indirect schema mappings. *Information Systems*, 31(8):697–732, December 2006.

[ZBC04] R. Zanibbi, D. Blostein, and J.R. Cordy. A survey of table recognition: Models, observations, transformations, and inferences. *International Journal of Document Analysis and Recognition*, 7(1):1–16, 2004.