Theoretical Foundations for Enabling a Web of Knowledge

David W. Embley* and Andrew Zitzelberger*

Brigham Young University, Provo, Utah 84602, U.S.A.

Abstract. The current web is a web of linked pages. Frustrated users search for facts by guessing which keywords or keyword phrases might lead them to pages where they can find facts. Can we make it possible for users to search directly for facts embedded in web pages? Instead of a web of human-readable pages containing machine-inaccessible facts, can the web be a web of machine-accessible facts superimposed over a web of human-readable pages? Ultimately, can the web be a web of knowledge that can provide direct answers to factual questions and support these answers by referencing and highlighting relevant base facts embedded in source pages? Answers to these questions call for distilling knowledge from the web's wealth of heterogeneous digital data into a web of knowledge. But how? Or, even more fundamentally, what, precisely, is this web of knowledge, and what is required to enable it? To answer these questions, we proffer a theoretical foundation for a web of knowledge: We formally define a computational view of knowledge in a way that enables practical construction and use of a web of knowledge.

1 Introduction

The web 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 stated facts with respect to ontologies. Annotating facts with respect to ontologies, 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 on the underlying web becomes a *Web of Knowledge*, a *WoK*.[1]

* Supported in part by the National Science Foundation under Grant #0414644.

[1] To many, this vision of a WoK constitutes the semantic web [37].

Although the 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. Thus, all constituent components for a WoK are industry 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.

Extraction ontologies provide a way to solve both creation and usage problems [13]. But, what exactly are extraction ontologies, and how do they resolve creation and usage problems? We answer these questions in this paper by formalizing extraction ontologies (Section 2), formalizing the notion of a WoK (Section 3), formalizing WoK construction procedures (Section 4), and formalizing user-friendly, WoK query-processing procedures (Section 5).

The success of the WoK vision depends on a solid theoretical foundation. Thus, the contributions of this paper are: (1) the formalization of extraction ontologies, knowledge bundles, and interconnected knowledge bundles as a WoK; (2) the formalization of WoK construction tools; and (3) the formalization of WoK usage tools. Indirectly, further contributions include: (1) the basis for an open architecture for pluggable tools for realizing a WoK along with implemented examples of construction and usage tools; and (2) the identification of strengths and weaknesses of WoK construction and usage tools and thus the identification of where opportunities lie for future research and development.

2 Extraction Ontologies

We base our foundational conceptualization for a web of knowledge on the conceptual modeling language OSM (Object-oriented Systems Model) [17]. 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 [4].

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.*[2]

[2] In forthcoming definitions, lexical predicates will be restricted to literal domain-value substitutions like strings, integers, phone numbers, email addresses, and account

– *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 objects* 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 an 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: meronym-holonym relationship sets grouped as an aggregation in an is-part-of hierarchy.* □

Example 1. Figure 1 shows an OSM-O model instance. Rectangular boxes are object sets—dashed if lexical and solid if non-lexical. Lines between object sets denote relationship sets. Participation constraints are next to relationship-set/object-set connections in a *min:max* format. (We use *1* as shorthand for *1:1*.) A white triangle denotes generalization/specialization with the generalization connected to the apex of the triangle and the specializations connected to the base. A black triangle denotes aggregation with the super-part connected to the apex of the triangle and the sub-parts connected to the base. Although graphical in appearance, an OSM-O diagram is merely a two-dimensional rendition of predicates and closed formulas as defined in Definition 1. □

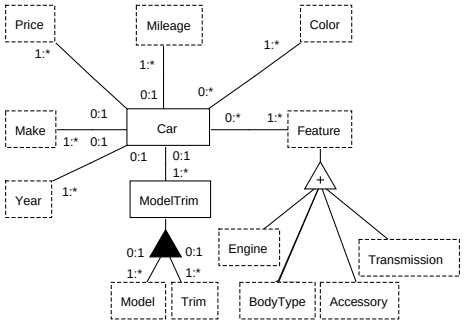


Fig. 1. OSM-O Model Instance.

Definition 2. *Let M = (O, R, C) be an OSM-O model instance. Let I be an interpretation for M that has a domain* $D = L_{ID} \cup O_{ID}$ ($L_{ID} \cap O_{ID} = \emptyset$) *and a declaration of* ***True*** *or* ***False*** *for each valid instantiation of each predicate in* $O \cup R$ *with values in D.*[3] *For predicates in O, valid instantiations require*

numbers. Non-lexical predicates will be restricted to substitutions of object identifiers that represent real-world objects like people, geopolitical entities, and buildings, or abstract concepts like a marriage or ownership.

[3] For an *O*-predicate, if a value instantiation *v* is **True**, *v* is in the *O* object set and is not in the *O* object set when the instantiation is **False**. Similarly, for an *R*-predicate,

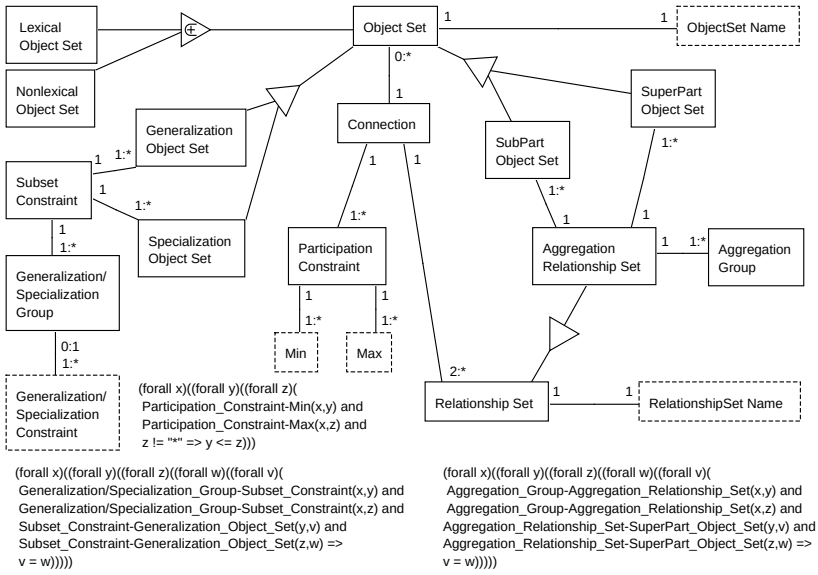


Fig. 2. OSM-O Meta-model.

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 1 contains facts about cars. A possible valid interpretation might include the facts $Car(Car_3), Year(2003), Car\text{-}Year(Car_3, 2003), Model(\text{“Accord”}), Trim(\text{“LX”}), ModelTrim(ModelTrim_{17}), Trim\text{-}isPartOf\text{-}ModelTrim(\text{“LX”}, ModelTrim_{17})$, and $Car\text{-}ModelTrim(Car_3, ModelTrim_{17})$. Note that the object sets Car and $ModelTrim$, being non-lexical, have identifiers for their domain-value substitutions. Constraints, such as $\forall x(Car(x) \Rightarrow \exists^{\leq 1} y(Car\text{-}Year(x, y)))$, all hold. □

The *OSM-O model instance* in Figure 2 is a meta-model that defines valid OSM-O model instances. Note that since OSM-O is predicate calculus (“disguised” in graphical notation), the three additional closed formulas in Figure 2 are appropriate well-formed-formula additions to the statements in the meta-model. Two of them ensure that the group members of any generalization/specialization and any aggregation have the same parent. The third ensures that min-cardinality values are no greater than max-cardinality values.

if an n-tuple instantiation $<v_1, ..., v_n>$ is **True**, $<v_1, ..., v_n>$ is in the R relationship set and is not in the relationship set when the instantiation is **False**.

Definition 3. *If M is a valid interpretation for the OSM-O meta-model instance, then M is a* valid *model instance.* □

Example 3. The car ontology in Figure 1 is a valid interpretation of the OSM-O meta-model in Figure 2. Part of the interpretation would include $ObjectSet\_Name$("Feature"), and if the $Feature$ object set has the identifier O_{13}, then also $Object\_Set(O_{13})$, $Generalization\_Set(O_{13})$, $Lexical\_Object\_Set(O_{13})$, and $Object\_Set\text{-}ObjectSet\_Name(O_{13}$, "Feature"). Note that the meta-model does not dictate names for object and relationship sets. We want them to be mnemonic, of course, as they are here and in Example 2, but we can make them mnemonic in any way we wish. This also solves the problem of naming a second binary relationship set between the same two object sets. In Figure 1, for example we could have two relationship sets between Car and $Price$: one of the predicate names could be $Car\_sellsFor\_Price$ while the other could be $Car\_hasManufacturerSuggested\_Price$. □

Theorem 1. *OSM-O is decidable.* □

Proof. Let $M = (O, R, C)$ be an OSM-O model instance. We show how to translate each component of M to an OWL-DL instance Z. Then, since OWL-DL is decidable, OSM-O is decidable.

- For each object set $S \in O$ create a class (owl:Class) in Z.
- For each lexical object set $S \in O$ create a data-type property (owl:DatatypeProperty) in Z with domain S and with the range being any appropriate data type.
- For each generalization/specialization group H with G as the generalization object set and S_1, ..., S_n as the specialization object sets, process constraints in C as follows:
  - Make each class created from S_1, ..., S_n a subclass of (rdfs:subClassOf) the class created for G.
  - If H has a union constraint, make the class created for G the union of (owl:unionOf) the classes S_1, ..., S_n.
  - If H has a disjoint constraint, make S_i disjoint with (owl:disjointWith) S_j, $(1 \leq i, j \leq n; i \neq j)$.
- Process relationship sets in R as follows:
  - For each n-ary relationship set $Q \in R$ with $n > 2$, for purposes of the translation, replace Q with a non-lexical object set S and binary relationships to each participating object set of Q. Add participation constraints *1* (= *1:1*) for each S-connection and let the participation constraints for connections to other object sets be as originally specified in Q. Create a class for S in Z.
  - (After translating n-ary relationship sets, $n > 2$, to binary relationship sets, all relationship sets in R are binary. These relationship sets include the binary relationship sets originally in R, the binary relationship sets created from n-ary relationship sets for purposes of translation, and all aggregation relationship sets.) For each relationship set in R, create an object property (owl:ObjectProperty) and a second object property and make eacn an inverse of (owl:inverseOf) the other.

- For each participation constraint in C, create a restriction with min and max constraints (owl:Restriction, owl:minCardinality, and owl:maxCardinality) for the appropriate object property created from R. □

We have implemented the transformation in the proof of Theorem 1.[4] Thus, when we discuss mappings to OSM-O (below), mappings to OWL-DL are implied and well defined. Further, it should be clear, that given a mapping of a model instance M to OWL-DL, the mapping of the instance data in M to RDF such that it corresponds to the generated OWL-DL is straightforward. Thus, when instance data is part of a mapping to an OSM-O model instance, a mapping of the data to RDF is also implied and well defined. We have also implemented this transformation to RDF.

Similar to the work by Buitelaar, et al. [8], we now show how to linguistically ground OSM-O. For us, 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. We then add linguistic recognizers for instance values, operators, operator parameters, and relationships.

Definition 4. *An* abstract data type *is a pair (V, O) where V is a set of values and O is a set of operations.* □

Definition 5. *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 an instance recognizer that identifies lexical patterns denoting values in V.*
3. *The operations set O includes two (additional) operators for each lexical object set: (1) an input operator to convert identified instances to the internal representation for V and (2) an output operator to convert instances in V to strings.*
4. *Each operation o in O has an operator recognizer that identifies lexical patterns as indicators that o applies. Further, the recognizer identifies lexical patterns that, along with instance recognizers, identify parameters for o.* □

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.[5] To be complete and precise, however, we give examples to illustrate how we have implemented them.

[4] Although unspecified in our proof, our implementation automatically assigns names to object and relationship sets based on given object-set names and relationship-set connections among names.

[5] Many kinds of recognizers are possible. Much work has been done on named entity recognition, and entire communities and conferences/workshops are devoted to these issues. We want the WoK vision to be able to include and make use of this work.

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. 3. Data Frames.

Example 4. Figure 3 shows two partial data frames for object sets in the OSM-O model instance in Figure 1, one for *Price* and one for *Make*. The *Price* data frame uses regular expressions for its recognizers, whereas the *Make* data frame uses lexicons. In our implementation, we can use either or both together. The *Price* data frame also shows a recognizer for an operator. The *p2* within curly braces indicates the expected appearance of the *Price* parameter *p2*. Thus a phrase like "under 15 grand" is recognized as indicating that the price of the car—parameter *p1*—should be less than $15,000. □

A data frame for a non-lexical object set is typically degenerate: Its value set is a set of object identifiers. Its operation set consists only of operators that add and remove object identifiers. Its name and synonyms and its recognizers that identify object instances, however, can be quite rich.

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 5. Suppose the *Car* object set in Figure 1 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*}. □

Definition 6. *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.* □

Definition 7. *An ontology is* linguistically grounded *if it can both "read" and "write" in some natural language. An ontology can "read" and "write" in some natural language if each object set and relationship set has a data frame.* □

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 [16]), but it is clear that opportunities abound for further research and develop-

ment. 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* $Car(Car_{734})$). Making the written description pleasing is, of course, more difficult.

Theorem 2. *An OSM-EO model instance is linguistically grounded.* □

Proof. Clear from Definitions 6 and 7. □

3 Web of Knowledge

Ontology is the study of "the nature of existence." Epistemology is the study of "the origin, nature, methods, and limits of human knowledge." In the previous section we have given a computational view of ontology—a view that lets us work with ontologies in information systems. Here we similarly give a computational view of epistemology. This computational view of epistemology constitutes the formal foundation for a web of knowledge.

Definition 8. *The collection of facts in an OSM-O model instance constitutes the* extensional knowledge *of the OSM-O model instance. The collection of implied facts derived from the extensional knowledge by inference rules constitutes the* intentional knowledge*. The extensional and intentional 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" [26]. "Knowledge" without some sort of truth authentication can be confusing and misleading.

Definition 9. Truth *is knowledge of things as they are, as they were, and as they are to come. "Knowledge of things" means facts about objects. By adding time intervals to facts, "as they are" means that the time interval covers the present; "as they were" means at some time interval in the past; and "as they are to come" means at some time interval in the future. If a fact holds over all time intervals (past, present, and future), the fact is an* eternal truth*.* □

Unfortunately, truth, without a perfect oracle, is always suspect. Believing axiomatically that it is impossible to computationally construct a perfect oracle, how can we compensate? We see three possibilities: (1) truth as community agreement—e.g., Wikipedia style; (2) probabilistic truth; and (3) truth derived from proper reasoning chains grounded in original sources. All three, unfortunately, are problematic: community agreement depends on the willingness of individuals to participate and to agree; probabilistic truth depends on establishing probabilities and on being able to derive probabilities for answers to queries—hard problems that do not scale well [12]; and reasoning with rules and fact sources depends on acceptance of the rules and fact sources as genuine.

For our vision of a WoK, we attempt to establish truth via provenance and authentication. We provide for reasoning with rules[6] and for ground facts in sources. We cannot, however, guarantee that rules and facts in sources are genuine. We thus compensate by simply exposing them. 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. We also ignore time intervals for now, leaving the time over which facts hold to be determined by understanding the original sources, the rules, and the context of the sources and rules.

Definition 10. *A* knowledge bundle *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 rule set where each rule is a horn clause whose body-predicates are either predicates in O or are head-predicates of other rules in R.* □

Definition 11. *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 set of real-world objects as object set y in knowledge bundle B_2.* □

4 WoK Construction

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

Definition 12. *A* transformation *is a 5-tuple (R, S, T, Σ, Π), 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 declarative source-to-target transformation statements, and Π is a set of procedural source-to-target transformation statements.* □

[6] In our implementation, we use Pellet [30].

Definition 12 leaves several of its components open—to take on specific meanings in a variety of knowledge-bundle building tools. The "set of resources" is undefined, but we intend this to mean resources such as WordNet and a data-frame library. "Target conceptualizations" are knowledge bundles. "Source conceptualizations" depend on sources whose fact conceptualizations can be formal, semi-formal, or informal. "Declarative" and "procedural" "source-to-target transformation statements" can be written in a variety of formal languages. Although leaving transformation statements loosely defined, as we do, presents some problems, it also presents some opportunities. Successfully developing automatic and good semi-automatic transformations over a broad spectrum of documents for a variety of ontological contexts should be both beneficial and profitable.

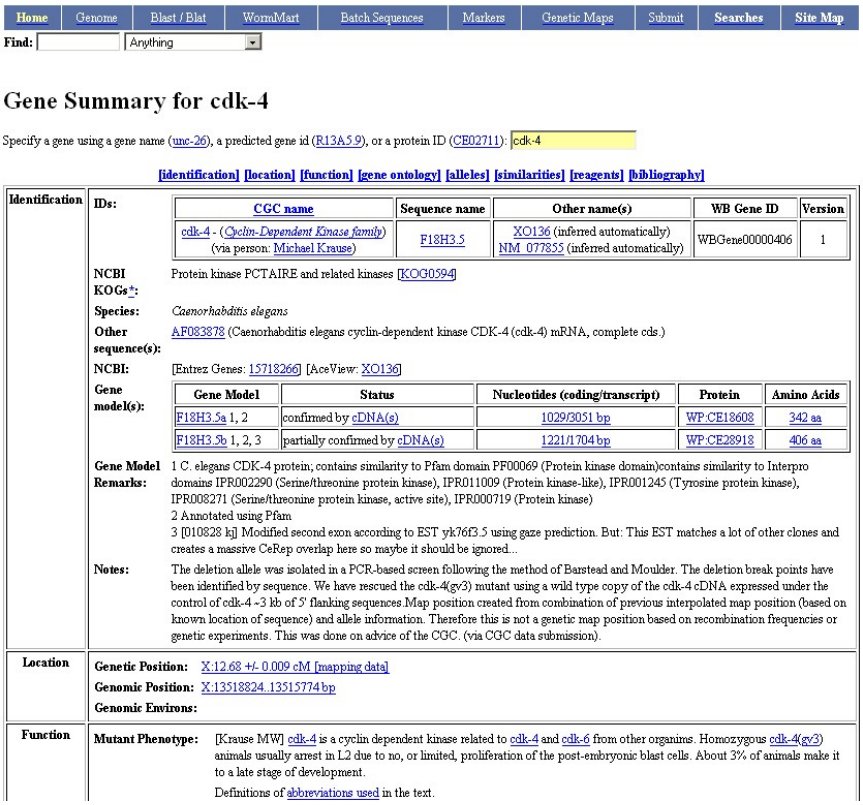
To be specific about some of the possibilities, however, we provide some examples. We first give some restrictive transformations guaranteed to capture all facts in the source and then mention some less restrictive transformations.

Definition 13. *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 $C_T \Rightarrow C_S$.* □

Theorem 3. *Let S be a nested table with a single label path to each data item, and let T be an OSM-O model instance. A transformation from S to T exists that preserves information and constraints.* □

Proof. (*sketch*) We show how to construct and populate T from S such that an inverse procedure exists and such that the constraints of T imply the constraints of S. (The transformation of the nested table in Figure 4 to the OSM-O model instance in Figure 5 illustrates the transformation.) To initialize T, we create a non-lexical object set to represent the table as a concept in the OSM-O ontology and label it "Table". Each label in S becomes an object set in T. Because of the restriction of S having a single label path to each data item, only the leaf nodes of T are lexical object sets (containing the data items). Relationship sets mark the path between labels down to the data items. Participation constraints on the parent sides in the tree of T are inferred from the number of items in the object sets of S and are assumed to be *1:** on the child sides. The reverse transformation is straightforward as the ontology is a hierarchical view of the table starting at the "Table" object set. The parent participation constraints of T imply the inferred participation constraints of S. □

We can exploit the direct correspondence between the nested tables of Theorem 3 and OSM-O in several ways. First, as intended, we can convert facts in nested tables into knowledge bundles for a WoK. Second, nested tables from the WormBase site are all sibling tables—they have identical, or nearly identical structure. We have shown elsewhere that we can automatically construct an ontology for the site (and any other site with sibling tables) and extract



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

Find: Anything

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. 4. Nested Table in a Molecular-Biology Web Page.

the information in all the tables to populate the ontology [32]. Third we can create extraction ontologies automatically (although they likely need some enhancement) [32]. Fourth we can turn the process around and let users specify ontologies via nested forms [33].

Theorem 4. *Let S be a relational database with its schema restricted as follows: (1) the only declared constraints are single-attribute primary key constraints and single-attribute foreign-key constraints, (2) every relation schema has a primary key, (3) all foreign keys reference only primary keys and have the same name as the primary key they reference, (4) except for attributes referencing foreign keys, all attribute names are unique throughout the entire database schema, (5) all relation schemas are in 3NF. Let T be an OSM-O model instance. A transformation from S to T exists that preserves information and constraints.* □

Proof. (*sketch*) For every relation, create a non-lexical object set and make its name the name of the relation. Except for attributes that reference foreign keys, create a lexical object set for each attribute and make its name the attribute name. For each schema S, create a binary relationship set between the non-lexical object set created for S and each lexical object set created from attributes in S.

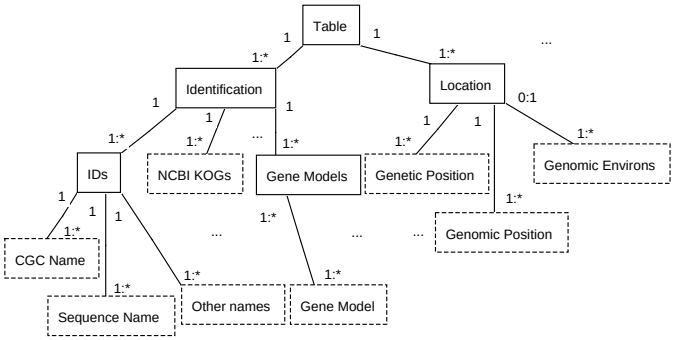


Fig. 5. Generated OSM-O Model Instance.

The participation constraint on the non-lexical side of all these relationship sets is *1*. For lexical object sets corresponding to primary-key attributes the participation constraints on the lexical sides are also *1*; otherwise the participation constraints on the lexical sides are *1:**. For each primary-key foreign-key reference, create a generalization/specialization between the schema's non-lexical object sets with the object set of the referenced key as the generalization and the object set of the referencing key as the specialization. For all other foreign-key references, create a binary relationship set between the non-lexical object sets of the two schema's with *0:1* as the participation constraint on the referencing side and *1:** as the participation constraint on the referenced side. The reverse transformation is a standard transformation of an OSM model instance to a relational database [15]. Since the standard transformation generates all constraints of S, the constraints of T imply the constraints of S. □

Theorem 4 is overly restrictive for most practical applications, but its restrictions allow us to illustrate the process well in a short description. Recently, Tirmizi et al. defined a transformation from a relational database directly to OWL that has fewer restrictions and that also preserves information and constraints [35]. Further, Tirmizi et al. write their transformation declaratively, using only Prolog-like rules for specifying their transformation, and thus use only the S, T, and Σ components of a general transformation (see Definition 12).

For a WoK, preservation and transformation requirements need not be so strict. We only need to be able to extract each base fact and represent it in an ontology. To make a WoK highly meaningful, however, we should recover as much as is possible of the underlying semantics—the facts, the constraints, and the linguistic connections. Therein lies the difficulty: some of the underlying semantics in source conceptualizations exist only implicitly and are thus difficult to capture, and some of the underlying semantics do not exist at all, having been discarded in the abstraction process of producing the conceptualization. But therein also lies the research opportunities. Many researchers are endeavoring to create automatic and semi-automatic procedures to capture richer semantics—making extensive use of the Π component of Definition 12. And some are making use of R component of Definition 12 to recover semantics lost in the abstraction

process. In general, there is an effort to recover as much of the semantics as possible from many different source genres. For example, researchers have investigated semantic recovery from relational databases [5, 6], XML [1, 38], human-readable tables [24, 25, 34], forms [22, 31], and free-running text [10].

For the last part of WoK creation—creating links among knowledge bundles—we rely on *record linkage* and *schema mapping*, also called *ontology alignment* or *ontology matching*. *Record linkage* is the task of finding entries that refer to the same entity in two or more data sources, and and *ontology matching* is the task of finding the semantic correspondences between elements of two ontology schemas. An extensive body of work exists for both record linkage [14] and ontology matching [21]. Unfortunately, both problems are extremely hard. General solutions do not exist and may never exist. However, "best-effort" methods do exist and perform reasonably well. For the WoK vision, we can use these best-effort methods to initialize and update *same-as* links for object identity and *equivalence-class* links for concept identity. Using "best-effort" methods in a "pay-as-you-go" fashion appears to be the best way to enable a WoK.

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 the 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 knowledge bundle.

Definition 14. *Let S be a source conceptualization and let T be a target conceptualization formalized as an OSM-EO. 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.* □

Observe that although Definition 14 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 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 for every predicate in S and if there is an operator in a data frame for every operator in S.

Applications of understanding include free-form query processing, advanced form-query processing, knowledge augmentation, and knowledge-bundle building for research studies.

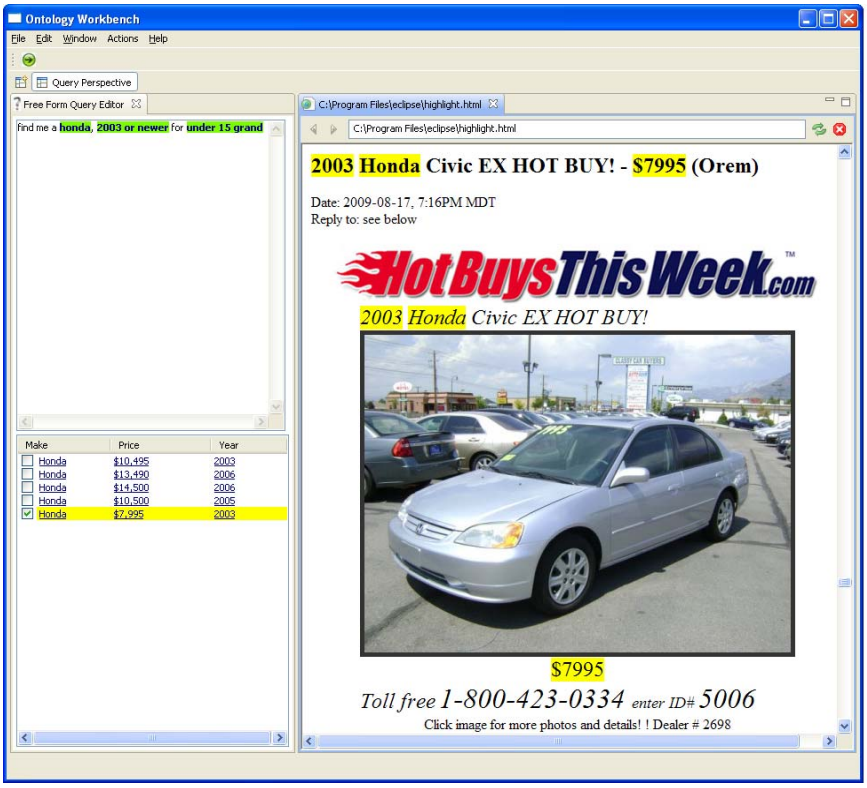


Fig. 6. Screenshot of WoK Prototype Showing Free-Form Query Processing.

Example 6. Free-Form Query Processing: Figure 6 illustrates free-form query processing within our WoK prototype. To "understand" a user query, our WoK prototype first determines which OSM-EO applies to the query by seeing which one recognizes the most instances, predicates, and operators in the query request. Having chosen the *Car* extraction ontology illustrated in Figures 1 and 3, the WoK applies the S-to-T transformation highlighting what it understands ("Find me a honda, 2003 or newer for under 15 grand"). Figure 7 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 in the lower-left box in Figure 6, 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 6 shows several highlighted values, which happens when the user checks one or more check-boxes before clicking. □

Example 7. Advanced Form-Query Processing The form in Figure 7 is for an alerter system for craigslist.org, which we have implemented. We use it as feedback to the user that the query has been understood. As such, it illustrates not only the ability of an OSM-EO to read and understand, but also its ability to write. Note, for example the conversion of "15 grand" to "$15,000" as well as the mnemonic names for predicates and operations. Besides providing feedback, this writing ability also lets the user know what else the OSM-EO knows about. A user U then has the opportunity to adjust the query or add additional constraints. For example, U may wish to also know if Toyotas are listed so long as 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, the user might wish to limit prices with $Between(Car.Price, \$11K, \$16K)$. Since the OSM-EO has general recognizers for prices, U can enter them in any recognizable format.

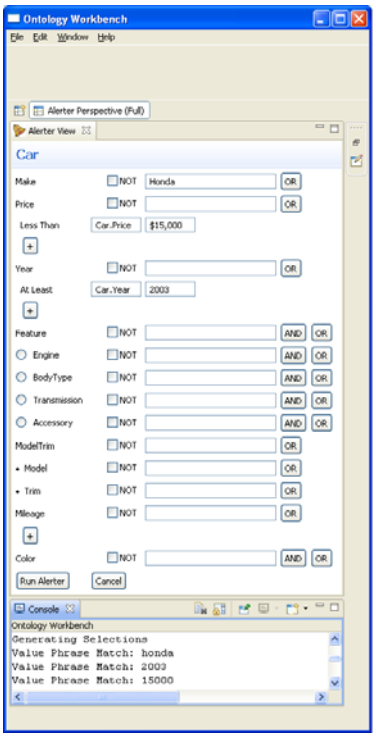


Fig. 7. Generated Form.

Example 8. Fact Finding 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 for them to be able to recognize facts in structured, semi-structured, and unstructured text. As an example, we can exploit this fact-finding ability to gather facts for a bio-research study and store them as a knowledge bundle for further analysis [19]. Gathering tasks for these research studies often takes 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.

Example 9. Knowledge Augmentation: Partial "understanding" is also useful. If two OSM-EO model instances, M_1 and M_2, can partially "understand" each other, they can be merged based on their common "understanding." M_1 can be seen as having been augmented by the part of M_2 that it did not already "understand" and vice versa. We can exploit "partial understanding" along with reverse engineering of semi-structured sources to grow ontologies. In our TANGO project, for example, we take a collection of ordinary tables with overlapping

information, reverse engineer them one at a time creating an OSM-EO model instance for each and then merge each into a growing ontology that represents the entire collection [34].

6 Conclusion

We have defined a formal foundation for a web of knowledge (a WoK). We have formalized fundamental WoK components: ontologies (OSM-O) in terms of decidable first-order logic and extraction ontologies (OSM-EO) linguistically grounded via data-frame recognizers. We have also formalized a computational view of a WoK that includes knowledge as a collection of facts embedded in ontologies, truth as knowledge bound to source documents for provenance and authentication, and knowledge bundles as consisting of OSM-EO model instances with valid interpretations super-imposed over source documents. In terms of this formal foundation, a WoK is a collection of knowledge bundles interconnected with links that bind together objects with the same identity and concepts with the same meaning.

Further, we have formally 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 shown that some source conceptualizations (e.g., specialized, but commonly occurring, nested tables, nested forms, and relational databases) have transformations 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) require best-effort, pay-as-you-go methods.

Finally, we have formally 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. Applications of understanding include free-form query processing, form-based query processing, fact finding for research studies, and knowledge augmentation. The purpose of an OSM-EO model instance is to establish understanding. How well OSM-EO model instances "understand" depends on how well they are able to "read" and "write" which, like WoK-construction, requires best-effort, pay-as-you-go methods.

Best-effort methods open the door to interesting and exciting research possibilities. We have implemented a WoK prototype [18] including some prototypical extraction ontologies [16]. We have also done some work on automated extraction-ontology construction [32, 24, 33, 34] and some work on free-form query processing [36, 2]. We nevertheless still have much work to do, even on fundamental WoK components such as creating a sharable data-frame library, constructing molecular-size ontology snippets, finding ways to more easily produce instance recognizers, reverse-engineering of many genres of semi-structured sources to 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 [20], best-effort information extraction [29], C-PANKOW [11], Q/A systems [27], bootstrapping pay-as-you-go data integration [28], large-scale deduplication [3], and OpenDMAP [23].

These collective efforts will eventually lead to a WoK—a realization of ideas of visionaries from Bush [9] to Berners-Lee [7] and Weikum [39]. Establishing a framework for a WoK by formalizing its basic components and establishing a firm theoretical foundation as we have done here can help enable this new kind of information system—a web of knowledge.

Acknowledgements: We would like to thank Cui Tao and Yihong Ding for coding the transformations of nested tables to OSM-EO model instances, Oliver Nina and Meher Shaikh for coding the craigslist.org alerter, Stephen W. Liddle for coding the SPARQL query augmenter and source highlighter, and many former and current students who have worked on our extraction-ontology engine.

References

1. R. Al-Kamha. *Conceptual XML for Systems Analysis*. PhD dissertation, Brigham Young University, Department of Computer Science, June 2007.
2. M. Al-Mumammed and D.W. Embley. Ontology-based constraint recognition for free-form service requests. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE'07)*, pages 366–375, Istanbul, Turkey, April 2007.
3. 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.
4. Aristotle. *Metaphysics*. Oxford University Press, New York, about 350 BC (1993 translation).
5. I. Astrova. Reverse engineering of relational databases to ontologies. In *Proceedings of the First European Semantic Web Symposium*, pages 327–341, Heraklion, Crete, Greece, May 2004.
6. J. Atoum, D. Bader, and A. Awajan. Mining functional dependency from relational databases using equivalent classes and minimal cover. *Journal of Computer Science*, 4(6):421–426, 2008.
7. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 36(25):34–43, May 2001.
8. 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.
9. V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, July 1945.
10. P. Cimiano. *Ontology Learning and Population from Text: Algorithm, Evaluation and Applications*. Springer Verlag, New York, New York, 2006.
11. 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.

12. N. Dalvi, C. Ré, and Dan Suciu. Probabilistic databases: Diamonds in the dirt. *Communications of the ACM*, 52(7):86–94, July 2009.
13. Y. Ding, D.W. Embley, and S.W. Liddle. Automatic creation and simplified querying of semantic web content: An approach based on information-extraction ontologies. In *Proceedings of the First Asian Semantic Web Conference (ASWC'06)*, pages 400–414, Beijing, China, September 2006.
14. 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.
15. D.W. Embley. *Object Database Development: Concepts and Principles*. Addison-Wesley, Reading, Massachusetts, 1998.
16. 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.
17. 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.
18. 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*, pages 532–533, Barcelona, Spain, October 2008.
19. 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. (in press).
20. 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.
21. J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag, Heidelberg, Germany, 2007.
22. A. Gal, G.A. Modica, and H.M. Jamil. Ontobuilder: Fully automatic extraction and consolidation of ontologies from web sources. In *Proceedings of the 20th International Conference on Data Engineering*, page 853, Boston, Massachusetts, March/April 2004.
23. 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.
24. 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.
25. 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.
26. Plato. *Theaetetus*. BiblioBazaar, LLC, Charleston, South Carolina, about 360BC. (translated by Benjamin Jowett).
27. D. Roussinov, W. Fan, and J. Robles-Flores. Beyond keywords: Automated question answering on the web. *Communications of the ACM*, 51(9), September 2008.

28. 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.
29. 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.
30. E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, and Y. Katz. A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2):51–53, March 2007.
31. W. Su, J. Wang, and F. Lochovsky. ODE: Ontology-assisted data extraction. *ACM Transactions on Database Systems*, 34(2):12.1–12.35, June 2009.
32. C. Tao and D.W. Embley. Automatic hidden-web table interpretation, conceptualization, and semantic annotation. *Data & Knowledge Engineering*, 68(7):683–703, July 2009.
33. 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)*, Gramado, Brazil, November 2009. (in press).
34. 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.
35. S. Tirmizi, J. Sequeda, and D. Miranker. Translating SQL applications to the semantic web. In *Database and Expert Systems Applications*, pages 450–464, Turin, Italy, September 2008.
36. M. Vickers. Ontology-based free-form query processing for the semantic web. Master's thesis, Brigham Young University, Provo, Utah, June 2006.
37. W3C (World Wide Web Consortium) *Semantic Web Activity Page*. http://www.w3.org/2001/sw/.
38. Y. Weidong, G. Ning, and S. Baile. Reverse engineering XML. In *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, volume 2, pages 447–454, Hangzhou, Zhejiang, China, June 2006.
39. 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.