Available online at www.sciencedirect.com

SCIENCE DIRECT®

Data & Knowledge Engineering 55 (2005) 84–102

DATA & KNOWLEDGE ENGINEERING

www.elsevier.com/locate/datak

A semiotic metrics suite for assessing the quality of ontologies

Andrew Burton-Jones [a], Veda C. Storey [a], Vijayan Sugumaran [b,*], Punit Ahluwalia [a]

[a] *J. Mack Robinson College of Business, Georgia State University, P.O. Box 4015, Atlanta, GA 30302, USA*
[b] *Department of Decision and Information Sciences, School of Business Administration, Oakland University, Rochester, MI 48309, USA*

Received 4 November 2004; accepted 4 November 2004
Available online 23 December 2004

Abstract

A suite of metrics is proposed to assess the quality of an ontology. Drawing upon semiotic theory, the metrics assess the syntactic, semantic, pragmatic, and social aspects of ontology quality. We operationalize the metrics and implement them in a prototype tool called the Ontology Auditor. An initial validation of the Ontology Auditor on the DARPA Agent Markup Language (DAML) library of domain ontologies indicates that the metrics are feasible and highlights the wide variation in quality among ontologies in the library. The contribution of the research is to provide a theory-based framework that developers can use to develop high quality ontologies and that applications can use to choose appropriate ontologies for a given task.

Keywords: Ontology; Quality metric; Semiotic theory; Ontology auditor

[*] Corresponding author. Tel.: +1 248 370 2831.

E-mail addresses: abjones@cis.gsu.edu (A. Burton-Jones), vstorey@cis.gsu.edu (V.C. Storey), sugumara@oakland.edu (V. Sugumaran), hispaa@cis.gsu.edu (P. Ahluwalia).

0169-023X/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2004.11.010

1. Introduction

Interpreting and reasoning with semantics remains a significant challenge in knowledge engineering. Over the last decade, one solution has been to use ontologies to serve as surrogates for the semantics of a domain. Borrowed from their role in philosophy where they serve as general descriptions of what can exist in the world [34], ontologies in knowledge engineering specify terms, relationships between terms, and inference rules for a topic [18]. Ontologies have been found to be useful in various applications, including information sharing among heterogeneous data sources [24], interpreting unstructured data on the World Wide Web [1], and creating and evaluating conceptual models [32].

The growth of ontology development has increased the need for research on principles for their creation, use, and evaluation [7]. The objective of this research is to present a suite of metrics that can be used to assess the quality of ontologies. The suite comprises of ten metrics derived from a theory of semiotics [30] that assess the syntactic, semantic, pragmatic, and social quality of an ontology. In this paper, we define and operationalize the metrics and implement them in an "Ontology Auditor". We then conduct an initial validation of the metrics by applying the ontology auditing tool to the DARPA Agent Markup Language (DAML) ontology library (http://www.daml.org/ontologies), a collection of 280 publicly available ontologies developed to support the Semantic Web. The contributions of the research are to (1) present a comprehensive and theory-based metrics suite that can support ontology creation and use; (2) show how such a metrics suite can be implemented in an ontology auditing tool; and (3) demonstrate the usefulness of the metrics by providing empirical evidence of the quality of ontologies in a widely used ontology library, namely the DAML library.

The paper is organized in five parts. Section 2 reviews recent research on ontology development, use, and evaluation. In Section 3, we propose and operationalize our metrics suite. Section 4 describes an implementation of the metrics suite in an Ontology Auditor. In Section 5, we conduct an initial validation of the metrics by applying them to the DAML ontology library. Section 6 concludes the paper and discusses avenues for future research.

2. Related research

Ontologies are used in knowledge engineering to enable two or more systems to *commit* to the meaning of terms [18]. By providing a formal, independent specification of ontological commitments, ontologies facilitate communication about a domain between systems or between humans and a system by allowing the parties in communication to resolve their views of the domain via the ontology [18].

Given the importance of ontologies in knowledge engineering, we might expect that there has been a significant body of research on ontologies. Some evidence supports this view. For example, recent reviews have noted the maturation of ontological research, especially methodologies, tools, algorithms, and languages for building ontologies [7,24]. There has also been significant research on applying ontologies in practice to test their usefulness [5].

Despite the substantial amount of research on ontologies, there remains relatively little research on approaches for evaluating ontologies [19]. Given that the essence of an ontology is the ontological commitments that it makes, knowledge engineers need a way to assess the quality of an ontology's commitments. Although such research is clearly important, two factors have made it difficult to conduct research on ontological quality.

First, although there has been a significant amount of research on ontology development, much of it has involved the construction of *new* methodologies, languages, and tools. As noted in [7], the proliferation of methodologies, languages, and tools has made it difficult for researchers to develop generic approaches for evaluating ontological quality that are independent of any one methodology, tool, or language. Reminiscent of the YAMA ("yet another modelling approach") syndrome in conceptual modelling, the proliferation of new approaches has hindered ontology evaluation. One potential solution, proposed by Corcho et al. [7], is to develop a common workbench for ontology development, but research on such a workbench is still at an early stage.

A second reason for slow progress in research on ontological quality is the difficulty of determining what *elements* of quality to evaluate. Two general approaches can be used to identify elements of ontological quality: induction and deduction. An inductive approach would involve empirically testing ontologies to identify characteristics of ontologies that are associated with favourable outcomes for an application. For example, Fox et al. [13] propose eight criteria for assessing the quality of ontologies that they found useful in their work: generality, competence, perspicuity, transformability, extensibility, granularity, scalability, and minimality. An advantage of the inductive approach is that the criteria that the researchers identify are known to be useful in at least one application context. The disadvantage is that it is difficult to establish that the identified criteria are generalizable to other contexts [34].

A deductive approach to identifying the relevant elements of quality would rely on a theory to derive relevant elements of ontological quality. For example, in conceptual modelling, researchers have used formal theories of ontology to define elements of conceptual modelling quality, e.g., Wand and Weber [33] and Chidamber and Kemerer [6] used Bunge's [2] formal ontology to evaluate various representation ontologies such as entity-relationship diagrams and object-oriented designs. Likewise, in knowledge engineering, Guarino and Welty [19] use formal ontology to evaluate the quality of *is–a* relationships in ontologies. The advantage of the deductive approach is that it provides a way to identify a parsimonious and interrelated set of metrics that are generalizable across contexts. The disadvantage is that the quality attributes that are deduced from the theory may not apply perfectly to any one ontology and it may be difficult to: (a) identify the best theory to use as a benchmark; and (b) ensure that the theory chosen is itself of high quality. For example, [34] suggests that there is insufficient evidence to determine the relative quality of different formal ontologies at this stage.

Despite the difficulties of studying ontological quality, several factors suggest that such research is critical. First, there is a growing reliance on ontologies in practice. For example, ontologies are central to the growth of the "Semantic Web" [1]. The Semantic Web is an extension of the current web, in which the semantics of terms found in web pages will be explicitly defined using online ontologies [1,9]. The aim of the Semantic Web is to create the infrastructure necessary for the web to become "machine-readable" so that agents can interpret and reason about semantics on web pages and, thus, perform complex, intelligent tasks [1]. Researchers are developing new languages and authoring tools to specify Web page semantics [12], determining how applications and intelligent agents might utilize the Semantic Web [22], and investigating approaches for querying the Semantic Web [21]. The ontological infrastructure to support the Semantic Web is being developed [5,23]. For example, the DARPA Agent Markup Language (DAML) ontology library contains 282 ontologies for this purpose. Because the Semantic Web relies so heavily on ontologies, the lack of high quality ontologies remains a significant barrier to its growth and feasibility [1,22].

A second reason for the importance of research on ontological quality is that the inherently decentralized nature of Web development suggests that there will be limited ability to centrally control ontological quality during development. Rather than having a limited number of comprehensive, high-quality ontologies (e.g., Cyc, [20]), experts suggest that many smaller, domain-specific ontologies will predominate [22,31]. As Hendler [22] predicts:

> *"The Semantic Web... will not primarily consist of neat ontologies that expert AI researchers have carefully constructed. I envision a complex Web of semantics ruled by the same sort of anarchy that rules the rest of the Web."*

The decentralized nature of ontology development would be less of a problem if there were an accepted methodology for ontology creation. However, there is currently no accepted way to develop ontologies [7,26]. Many ontology developers also fail to understand basic ontological relationships [19]. As a result, many existing ontologies are "muddle headed", "of wildly different quality", and "mutually inconsistent" [29]. The proliferation of low-quality ontologies is problematic. Agents that use ontologies containing incomplete, inaccurate, or misleading knowledge cannot perform tasks successfully. A poor quality ontology can reduce efficiency by requiring superfluous ontologies to be read and can reduce effectiveness by providing the agent with poor information. Metrics are needed to evaluate the quality of ontologies, to support their design, and inform their use. Although some metrics have been proposed for ontology evaluation [13,8,19], much work remains [34].

3. Metrics suite for evaluating domain ontologies

Weber [34] distinguishes between formal ontologies, used to describe reality in general, and material ontologies, used to describe specific aspects of reality. Material ontologies include:

- application ontologies—specify definitions needed for a particular application,
- domain ontologies—specify conceptualizations specific to a domain,
- generic ontologies—specify conceptualizations generic to several domains, and
- representation ontologies—specify conceptualizations that underlie knowledge representation formalisms (e.g., frames).

This section presents a metrics suite to evaluate the following material ontologies: generic, domain, and application ontologies. Our approach follows the measurement tradition in software engineering in which researchers attempt to identify the internal attributes of programs (e.g., coupling and cohesion) that give rise to external quality attributes (e.g., maintainability and performance) [10]. Similar to Chidamber and Kemerer's [6] metrics suite for object-oriented design, our aim is to develop a metrics suite for key *internal* attributes of ontologies, describe the metrics' theoretical basis, operationalize them, and collect data to test their initial feasibility. Our work focuses solely on internal attributes. Testing the relationship between these internal attributes

Table 1
Evaluation emphasis of the proposed metrics suite

Type of material ontology	Generic (independent of any application; low value to any one application)	Domain (medium independence from application; medium value in any one application)	Application (not independent from an application; highly valuable to the application)
Emphasis in metrics suite	High	High	Medium

and a range of external attributes of ontologies (e.g., maintainability, understandability, adoption, and application performance) is important, but outside the scope of this research.

Metric development and ontological engineering both share an important problem: metrics and concepts that are very useful in a specific context can be difficult to apply to other contexts, whereas metrics and concepts that are generally applicable may have limited usefulness in any one application [18,34]. Table 1 presents our approach to evaluating ontologies. The aim of our approach is to identify metrics that are general enough to apply to any application or domain, yet can still assess internal attributes relevant for specific applications.

To develop a metrics suite that is independent of any one application (Table 1), we followed a deductive, rather than an inductive, approach to identify the relevant elements of ontological quality. In prior research, researchers who have followed a deductive approach have generally used a formal ontology as the theoretical benchmark for identifying the relevant elements of quality [19,34]. While this approach has strengths, formal ontologies emphasize high-level, philosophical issues rather than pragmatic issues relevant for specific applications. To develop a more tailorable approach, this research adopts Stamper's et al. [30] semiotic framework, a general theoretical framework derived from linguistics that includes general elements of quality. It also explicitly includes pragmatic issues, enabling us to develop a metrics suite that is widely applicable yet can be tailored to the needs of specific applications.

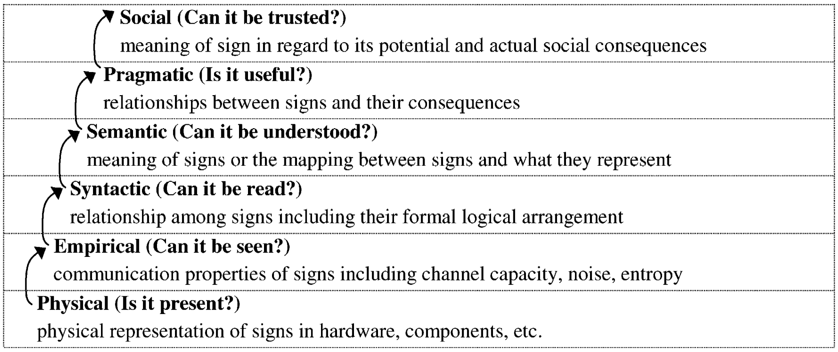
Semiotics studies the properties of signs. It assesses, for example, whether the sign used for "Chair" in an ontology is good or bad, clear or unclear. Ontologies use symbols, or signs, to describe terms. For example, a Computer Science ontology in the DAML library (http://www.daml.org/ontologies/64) includes:

```
<Class ID = "Chair">
   <label>chair</label>
   <subClassOf resource = "#AdministrativeStaff"/>
   <subClassOf resource = "#Professor"/>
</Class>
```

Several signs are manifest in this script. The terms "Class" and "subClassOf" are signs with meaning in the DARPA Agent Markup Language (DAML). The terms "Chair" and "Professor" are signs for things in the real world the ontology describes. Stamper et al. [30] provide a 6-level semiotic framework to support the analysis of signs, summarized in Table 2.

Table 3 proposes a suite of metrics for evaluating ontologies based upon the semiotic framework in Table 2. The suite consists of metrics for syntactic, semantic, pragmatic, and social quality. Metrics for physical and empirical quality are not included as they deal with implementation details.

Table 2
Semiotic framework



Social (Can it be trusted?) meaning of sign in regard to its potential and actual social consequences
Pragmatic (Is it useful?) relationships between signs and their consequences
Semantic (Can it be understood?) meaning of signs or the mapping between signs and what they represent
Syntactic (Can it be read?) relationship among signs including their formal logical arrangement
Empirical (Can it be seen?) communication properties of signs including channel capacity, noise, entropy
Physical (Is it present?) physical representation of signs in hardware, components, etc.

The arrows represent dependency relationships in which A → B reflects that B depends on A.

Table 3
Proposed metric suite for ontological auditing

Overall metric	Metrics suite	Attributes	Description
Ontology quality	Syntactic quality	Lawfulness	Correctness of syntax
		Richness	Breadth of syntax used
	Semantic quality	Interpretability	Meaningfulness of terms
		Consistency	Consistency of meaning of terms
		Clarity	Average number of word senses
	Pragmatic quality	Comprehensiveness	Number of classes and properties
		Accuracy	Accuracy of information
		Relevance	Relevance of information for a task
	Social quality	Authority	Extent to which other ontologies rely on it
		History	Number of times ontology has been used

Our proposed metrics suite is guided by three design principles. First, the metrics suite should not be limited to a particular type of user. It can support people who assess ontological quality (e.g., software auditors and ontology developers) as well as machines (e.g., virtual software auditors and ontology development environments), but it is independent of the capabilities of a specific person or machine. Second, the metrics suite is not limited to particular types of ontologies. It should support audits of single ontologies on their own, ontologies in the presence of a library of others, and ontologies of different languages and domains. Finally, the metrics suite is designed to be comprehensive, yet parsimonious.

The metrics suite is also guided by two assumptions. First, the metrics assume that the ontology is written in a known language. This is defensible given the increasing standardization of ontology languages [22]. Second, the metrics assume that there is an independent body of semantics that can be used to assess an ontology's semantics. To the extent that the ontology is the only known description of some domain, only a portion of the metrics could be used.

As shown in Table 2, the semiotic framework is multidimensional. Each metric can be rated using a continuous scale (not a categorical yes/no variable). Each metric also depends on its preceding metric for its applicability (e.g., semantic quality depends on syntactic quality). In measurement theoretic terms, ontology quality in Table 3 is a *formative* rather than a *reflective* construct [14]. In reflective constructs, the overall construct (ontology quality) would be equally reflected in each of its subconstructs and measures. All elements, therefore, would highly correlate. In formative constructs, the overall construct is *formed* by its subconstructs and measures. Thus, one metric (e.g., semantic quality) can be, but need not be, highly correlated with another metric (e.g., pragmatic quality). Although each metric is considered important, and could be equally weighted, the weights for specific metrics and attributes could be allowed to vary across application scenarios. This is the benefit of formative constructs: they allow one to develop general measurement frameworks that can be tailored for specific contexts.

Weights can be set in two ways: (1) by empirical evidence (empirically deriving the optimal weight of each component for predicting an external quality attribute such as application performance within a specific context; or (2) by expectation, (setting the weights automatically based on past experience or from instruction by a user). In the latter approach, an ontology developer, for example, could give lower weight to the 'relevance' metric if he/she was developing a generic ontology or a user could give lower weight to the 'history' metric if he/she was studying a new domain. Once the weights have been determined, the metrics can be used to rate one or more ontologies (e.g., those in the DAML library). Applications can then use the evaluations to decide which ontologies to use. Likewise, ontology developers can use the metrics as design principles when building ontologies.

Table 4 operationalizes each metric. Two types of metrics are used: absolute and relative. Most of the metrics are absolute assessments in which the numerical value of a metric for an ontology varies between zero and one. For three metrics (comprehensiveness, authority, and history), the assessment is relative rather than absolute. The values for a given ontology will depend on an external benchmark such as the metric's average value across all the ontologies in the ontology library in which the ontology exists. As the numerical values of these relative scores could exceed one for any given ontology, the scores for these metrics are normalized so that the values of all metrics varies between zero and one prior to calculating overall ontological quality. As a formative construct [14], overall quality (Q) is a weighted function of its syntactic (S), semantic (E), pragmatic (P), and social (O) qualities (i.e., $Q = b_1 \times S + b_2 \times E + b_3 \times P + b_4 \times O$). The weights sum to unity. In the absence of pre-specified weights, the weights are assumed to be equal. We explain each family of metrics in turn.

Syntactic quality (S) measures the quality of the ontology according to the way it is written. Two metrics are used. *Lawfulness* is the degree to which an ontology language's rules have been complied. Not all ontology editors have error-checking capabilities; however, without correct syntax, the ontology cannot be read and used. *Richness* refers to the proportion of features in the ontology language that have been used in an ontology (e.g., whether it includes terms and axioms, or only terms). Richer ontologies are more valuable to the user (e.g., agent). Ongoing research is testing the value of adjusting this metric by the frequency of use of each feature (e.g., in accordance with Zipf's law).

Semantic quality (E) evaluates the meaning of terms in the ontology library. Three attributes are used: interpretability, consistency, and clarity. *Interpretability* refers to the meaning of terms

Table 4
Determination of metric values

Attributes	Determination
Overall quality (Q)	$Q = b_1 \cdot \mathrm{S} + b_2 \cdot E + b_3 \cdot P + b_4 \cdot O$
Syntactic quality (S)	$S = b_{s1} \cdot \mathrm{SL} + b_{s2} \cdot \mathrm{SR}$
Lawfulness (SL)	Let X be total syntactical rules. Let Xb be total breached rules. Let NS be the number of statements in the ontology. Then SL = Xb/NS
Richness (SR)	Let Y be the total syntactical features available in ontology language. Let Z be the total syntactical features used in this ontology. Then SR = Z/Y
Semantic quality (E)	$E = b_{e1} \cdot \mathrm{EI} + b_{e2} \cdot \mathrm{EC} + b_{e3} \cdot \mathrm{EA}$
Interpretability (EI)	Let C be the total number of terms used to define classes and properties in ontology. Let W be the number of terms that have a sense listed in WordNet. Then EI = W/C
Consistency (EC)	Let $I = 0$. Let C be the number of classes and properties in ontology. $\forall C_i$, if meaning in ontology is inconsistent, $I + 1$. Therefore, I = number of terms with inconsistent meaning. $E_C = I/C$
Clarity (EA)	Let C_i = name of class or property in ontology. $\forall C_i$, count A_i, (the number of word senses for that term in WordNet). Then EA = A/C
Pragmatic quality (P)	$P = b_{p1} \cdot \mathrm{PO} + b_{p2} \cdot \mathrm{PU} + b_{p3} \cdot \mathrm{PR}$
Comprehensiveness (PO)	Let C be the total number of classes and properties in ontology. Let V be the average value for C across entire library. Then PO = C/V
Accuracy (PU)	Let NS be the number of statements in ontology. Let F be the number of false statements. PU = F/NS. Requires evaluation by domain expert and/or truth maintenance system
Relevance (PR)	Let NS be the number of statements in the ontology. Let S be the type of syntax relevant to agent. Let R be the number of statements within NS that use S. PR = R/NS
Social quality (O)	$O = b_{o1} \cdot \mathrm{OT} + b_{o2} \cdot \mathrm{OH}$
Authority (OT)	Let an ontology in the library be OA. Let the set of other ontologies in the library be L. Let the total number of links from ontologies in L to OA be K. Let the average value for K across ontology library be V. Then OT = K/V
History (OH)	Let the total number of accesses to an ontology be A. Let the average value for A across ontology library be H. Then OH = A/H

(e.g., classes and properties) in the ontology. Preferably, the knowledge provided by the ontology can map into meaningful real world concepts. This is achieved by checking that the words used by the ontology exist in another independent semantic source, such as a domain-specific lexical database or a comprehensive, generic lexical database such as WordNet [11]. *Consistency* is whether terms have a consistent meaning in the ontology. For example, if an ontology claims that X is a subclass_of Y, and that Y is a property of X, then X and Y have inconsistent meanings and are of no semantic value. As Guarino and Welty [19] show, ontological terms such as *is–a* are often used inconsistently. *Clarity* is whether the context of terms is clear. For example, if an ontology claims that class "Chair" has the property "Salary", an agent must know that this describes academics, not furniture.

Pragmatic quality (P) refers to the ontology's usefulness for users or their agents, irrespective of syntax or semantics. Three criteria are used. *Accuracy* is whether the claims an ontology makes are "true". Although this is difficult to evaluate, it is an important element of ontological quality that can be assessed automatically (e.g., using learning mechanisms or truth maintenance systems) or by a person (e.g., a domain expert). *Comprehensiveness* is a measure of the size of the ontology.

Larger ontologies are more likely to be complete representations of their domains, and provide more knowledge to the agent. *Relevance* is whether the ontology satisfies the agent's specific requirements. This requires some knowledge of the agent's needs prior to evaluation. This metric is coarse because it checks whether the ontology contains the type of information the agent uses (e.g., property, subclass, etc.), rather than the actual semantics needed for specific tasks that the agent performs (e.g., properties or subclasses needed to interpret a piece of information).

Social quality (*O*) reflects the fact that agents and ontologies exist in communities. Two attributes of social quality are proposed. The *authority* of an ontology is the number of other ontologies that link to it (define their terms using its definitions). More authoritative ontologies signal that the knowledge they provide is accurate or useful. The *history* is the number of times the ontology is accessed. We assume that ontologies with longer histories are more dependable.

4. Implementation of ontology auditor agent

The metrics were implemented in an automated ontology auditor, whose architecture is shown in Fig. 1. We use the term "auditor" because we believe that such a tool has an important role in virtual market places that rely on ontologies, just as human auditors have an important role in traditional financial market places. The proposed auditor is an agent in that it operates autonomously to assess the goodness of an ontology before that ontology is used by an application. The auditor agent is comprised of three components: (a) search component, (b) rating component, and (c) publishing component. Applications can interface with the ontology auditor agent and can request it to evaluate ontologies in a particular domain. The auditor agent returns the scores for the ontologies so that the application can choose the appropriate ontologies to use.

The ontology auditor agent carries out a three-step process. First, the search component searches for ontologies in specified domains (e.g., the DARPA ontology library) based on their common ontology-language file extensions (e.g., file.daml.) Second, the rating component assesses each ontology using online sources of semantics (e.g., WordNet [28]) and rules for each metric. The rating component gives a rating for each metric and an overall average rating. It does not give a recommendation whether to use the ontology. This decision is left to the application using the information. Third, the publishing component publishes its assessment of the ontology in a designated location so that other agents can read it. The Ontology Auditor Agent has been

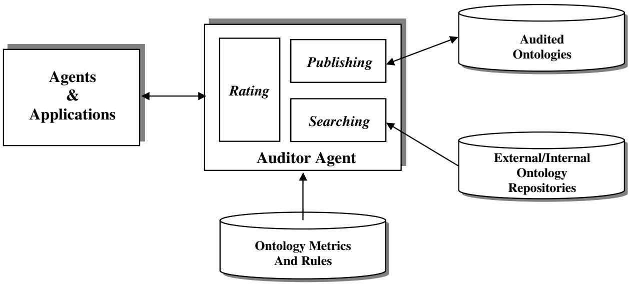


Fig. 1. Ontology Auditor Agent Architecture.

implemented in C++ and applied to the DAML ontologies. The auditor agent utilizes the WordNet [28] web service to determine word senses of terms. The agent also uses a knowledge base that contains the ontology metrics and rules to be used in evaluating ontologies.

The following sections outline how each metric was implemented in the ontology auditor, using examples from the DAML ontology library.

4.1. Search component

The search component continually evaluates ontologies. New ontologies from different libraries can be evaluated on demand; e.g., if an application requests knowledge on a domain that is not covered by the auditor's published list of evaluated ontologies. It contains meta-information about the ontologies and their domains.

4.2. Rating component

The rating component contains a module for each metric. The modules are described below.

4.2.1. Lawfulness module

Lawfulness is measured by searching for instances of incorrect syntax used within the ontology. The lawfulness module retrieves web pages containing ontologies (e.g., http://www.daml.org/ontologies/uri.html). Using a markup checker (e.g., http://www.daml.org/validator/), these pages are then parsed for syntactic errors and the number of errors detected reported. For example, applying this step to the Calendar ontology in the DAML ontology library (http://www.daml.org/ontologies/134) receives the following error:

ParseException: {E201} Syntax error when processing <EOF>. *Input to RDF parser ended prematurely.*

4.2.2. Richness module

The number of ontological properties used to describe each ontology provides a measure of richness. The score counts the number of different features used. The module determines how many features are used in each ontology (e.g., by importing data from a table of DAML features used in each ontology at http://www.daml.org/ontologies/features). For example, in the DAML ontology library, an ontology lacking in richness is the Instance ontology (http://www.daml.org/ontologies/77), which only uses two types of terms (subclass and type). In contrast, the Research Information ontology (http://www.daml.org/ontologies/221) contains 21 different types of terms, including class and subclass, property and subproperty, intersection, inverse, disjoint, domain, range, and cardinality.

4.2.3. Interpretability module

Interpretability is measured by checking WordNet to determine if the terms in an ontology are meaningful. The DAML pages are parsed for classes and properties. If the class names or property names are phrases (as is common in the DAML library; e.g., "LiquefiedGasCarrierWithTank OnDeck") these are modified before searching WordNet.

4.2.4. Consistency module

This module checks the internal consistency of ontologies. Inconsistencies occur when the same term is used in two or more ways within one ontology. For example, if term (*X*) is listed as a subclass of term (*Y*), it would be inconsistent if *X* also appeared as a super-class of *Y* elsewhere in the ontology. Similarly, if *X* is a property of *Y*, it should not also be a subclass of *Y*. Inconsistencies should be detected to avoid reaching incorrect inferences. For example, in the DAML profiling ontology (http://www.daml.org/ontologies/237) ''gender'' is listed as both a property and a class.

4.2.5. Clarity module

This is an extension of the interpretability module. Class and property names in WordNet are either single words (e.g., person) or phrases (e.g., firstName). The clarity metric checks for the number of senses in WordNet for the class or property name as a whole (whether a single word or phrase). Interpretability checks for the existence of the individual words (e.g., person, first, and name). An example of an 'unclear' word, for example, is the class ''break'' (found in the Agenda ontology http://www.daml.org/ontologies/238) which is highly polysemous, having 15 word senses in WordNet. Ideally, the ontology would use words with precise meanings (e.g., ''intermission'', which has only two senses) because automated approaches for resolving the context of polysemous words remains difficult [27], labor consuming [25], and unsatisfactory [15,16].

4.2.6. Comprehensiveness module

The total count of classes and properties in an ontology is reported. This is another extension of the interpretability module. As shown in Table 5, most DAML ontologies are small (1–20 terms). In contrast, some very large ontologies (e.g., Cyc, http://www.daml.org/ontologies/225) include over 2700 terms.

4.2.7. Accuracy module

This module tests whether knowledge given by the ontology is true. For example, in the DAML ontology library, the Computer Science ontology (http://www.daml.org/ontologies/225) states that 'staff' is a subclass of a department and, therefore, inherits the department's properties ('has_staff', 'has_courses', and 'has_URL'). This is inaccurate; staff are part of (not a subclass of) a department, and, therefore, should not inherit its properties. Accuracy is determined by checking knowledge in the ontology against existing knowledge known to be true. We are investigating how this could be performed automatically via a truth maintenance system for ontologies that provide axioms. Because many ontologies in the DAML ontology library do not provide axioms, the assessment of accuracy remains a manual process performed by a domain expert rather than an automatic step.

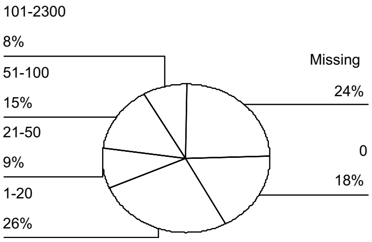
4.2.8. Relevance module

The relevance module examines the degree to which the ontology provides information of the type needed by an application. The module investigates four types of information that may be useful for different applications: class/subclass, property, cardinality, and a broad category called 'set knowledge' that includes restrictions, inverse, union, disjoint, complement, etc. To operationalize this step, we need to choose a particular application.

Table 5
Results from evaluation of DAML library[a]

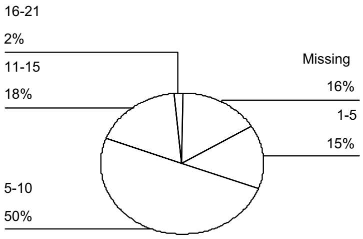
Syntax

Lawfulness



Values are the total number of syntax errors per ontology

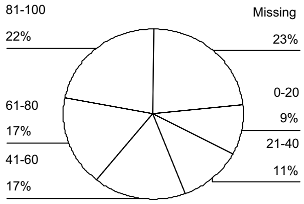
Richness



Values are the number of types of syntax used (max = 67)

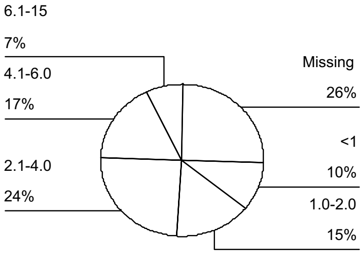
Semantics

Interpretability



Values are the percentage of words (used in class/property names) that exist in WordNet

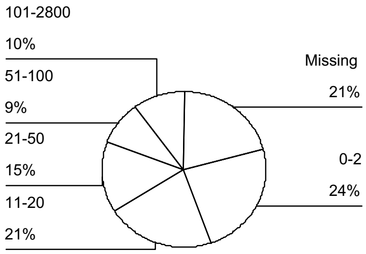
Clarity



Values are the average number of word senses for each ontology

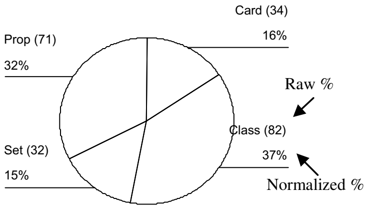
Pragmatics

Comprehensibility



Values are the sum of the classes and properties in each ontology

Relevance



Values are the percentage of ontologies that provides these semantics. Raw percentage = percentage of ontologies, normalised percentage = ratio out of 100%

[a] Missing: represents ontologies that were inaccessible when the evaluation was conducted (e.g., due to broken links).

In this research, we investigate the relevance of ontologies for *context aware query processing*. Our prior research [3,4] developed a Semantic Retrieval System (SRS) that implements a heuristics-based methodology for context-aware query processing on the Web. Following the tradition of query expansion [17,32], SRS expands a user's natural language query using lexically related terms (from WordNet) and domain-related terms (from the DAML ontology library), to contextualize the query so that it can obtain more relevant results. The effectiveness of SRS clearly depends on the ability of the DAML ontology library to provide relevant information. SRS primarily uses class/subclass relationships in the DAML library. Thus, in our research, the *relevance* module calculates the extent of class/subclass information provided by the ontology. For other applications, the module would calculate the extent to which the ontology provided the type of information needed by that application.

4.2.9. Authority module

Authority is the number of references made to one ontology from other ontologies. Applying this module to the DAML ontology library, the DAML page for each ontology is first parsed for references to other ontologies and the number of references to each ontology in the library is counted. An example of authority can be seen in the following definition of a class in the Computer Science ontology (http://www.daml.org/ontologies/65) that references "univ1.0.daml" for its definition of Faculty.

```
<Class ID = “Faculty”>
<equivalentToresource = “http://www.cs.umd.edu/projects/plus/DAML/onts/univ1.0.daml#Faculty”/>
```

4.2.10. History module

Implementing the history module requires either (a) the ontology library to publish the frequency of access to each ontology by other applications, or (b) an agent to track the number of times it uses each ontology. Many ontology libraries, including the DAML ontology library, do not provide information on the frequency of access to each ontology. In these cases, we would use the latter method to calculate history.

4.3. Publishing component

The results are stored in the Audited Ontologies database. The publishing component dynamically creates an html document that incorporates the assessment scores and related information using a predefined template. Human and/or software agents can access these pages to obtain the ratings and decide whether to use an ontology.

5. Analysis of DAML ontologies

To provide a test of the metrics' feasibility, the ontology auditor was applied to the DAML ontology library. In Table 5, we present evidence from testing six of the metrics. We have not yet conducted a full test of all the metrics against the DAML library. Therefore, the results in this section provide indicative but not definitive evidence for the usefulness of the metrics suite.

5.1. Syntax

The results for lawfulness in Table 5 indicate that only 18% of the DAML ontologies are free from syntax errors. Thirty-five percent of the ontologies contain 1–50 syntax errors, and over 20% contain greater than 50 errors. Although many ontologies had adequate syntax, the very poor quality of some ontologies reduced the overall quality in the library. As a proportion of the classes, properties, and instances per ontology, the average number of syntax errors per statement across the whole library was 1 (i.e., one syntax error per fact provided by the ontology). Improved tools would help prevent and detect syntax errors in these languages.

The results for richness show that over 60% of the ontologies use 1–10 different types of syntax and 20% use 10–20. DAML provides 67 types or 'features' of syntax. Some of these merely reflect language differences e.g., subclass available either using an RDF or DAML syntax. Nevertheless, 51 distinct types of syntax are available. None of the ontologies uses even half of the available syntax. This indicates that either the ontologies are underdeveloped, or many of the syntactical features are unnecessary.

5.2. Semantics

Almost 40% of the ontologies contained words that were clear (had between one and four senses on average). Nearly 25% of the ontologies, however, contained classes and properties that were highly polysemous, with an average of more than four senses for each class and property. Such ontologies could cause problems for agents needing knowledge about a specific sense of a term. Another 10% of the ontologies contained less than one sense per class or property name (i.e., included terms that had no meaning in WordNet).

The results for interpretability further showed that a surprisingly large number of ontologies contained meaningless terms. Only approximately 20% of the ontologies had more than 80% of their terms existing in WordNet. Across the entire library, less than half of the words in the library (43%) had meaning in WordNet. Overall, ontology designers appear to frequently use terms that do not exist in common English. An intuitive explanation is that these terms are merely highly domain-specific. While the auditor could be expanded to consult domain specific thesauri, our analysis found that many ontologies simply use acronyms, non-words, misspelled words, or non-English words. Tools could be developed to help ontology engineers use precise semantics when constructing ontologies.

5.3. Pragmatics

Although the DAML library contains some very large ontologies, most are small, with almost half containing only 1–20 terms (Table 5). Many represent knowledge about very narrow domains or capture small parts of larger domains, supporting the idea that agents will have to use these ontologies as a collective, rather than independent, source of domain knowledge [31]. SRS uses information on classes and subclasses. As shown in Table 5, over 80% of the ontologies provide such information. The DAML library is also relevant for applications that require information on properties of classes. The ontologies are less relevant for applications that require knowledge about cardinality or sets.

5.4. Total quality

Table 6 presents the analysis of overall quality. As described above, all metrics were weighted equally in the initial validation and the weights summed to one. Because not all metrics were tested, the numbers for overall quality should be considered indicative rather than definitive. Nevertheless, several observations can be made regarding the DAML ontologies as a whole. The mean values indicate that the average ontology has adequate syntax, interpretability, clarity, and relevance. Nevertheless, improvements can certainly be made. For the average ontology, 18% of its syntax is incorrect, 37% of its terms are uninterpretable, 22% of its terms are polysemous, and 18% of its statements are irrelevant (for applications that use subclass information). The mean values for richness and comprehensibility are much lower, indicating that DAML ontologies are generally small and unsophisticated. The range of values (low to high) is also instructive. The high ratings for some ontologies are promising, but the low values indicate that there is a clear need to improve the quality of many ontologies. Overall, the results provide strong empirical support for Hendler's [22] prediction that ontologies on the Semantic Web would be of widely different quality.

5.5. Implications of the metrics suite for ontology design

Although there is currently no standard ways to develop ontologies [7], our results suggest that such research is important. The metrics suite, therefore, could assist ontology developers in creating better designs. Knowledge and application of these metrics could help developers:

- capture a more comprehensive and consistent representation of their domain;
- check the syntax of their ontologies;
- ensure that the semantics they use are meaningful and precise; and
- develop an ontology so that it is relevant for many users/agents.

Table 6
Total quality

Metric	Dimension	Description	Low[a]	Mean[a]	High[a]
Syntax (S)	Lawfulness (SL)	Percentage of correct syntax per class and prop	0.00	0.82	1.00
	Richness (SR)	Percentage of available syntax used	0.04	0.17	0.41
	Total	$1/2SL + 1/2SR$	0.02	0.50	0.71
Semantics (E)	Interpretability (EI)	Percentage of words used that exist in WordNet	0.00	0.63	1.00
	Clarity[b] (EA)	Average precision of words in ontology	0.07	0.78	1.00
	Total	$1/2EI + 1/2EA$	0.04	0.71	1.00
Pragmatics (P)	Comprehensibility (PO)	Size as percentage of the largest (capped at 500)	0.00	0.11	1.00
	Relevance (for SRS) (PR)	Percentage providing subclass information	0.00	0.82	1.00
	Total	$1/2PO + 1/2PR$	0.00	0.47	1.00
Total		$1/3S + 1/3S + 1/3P$	0.02	0.56	0.90

[a] Low/Mean/High: represents the lowest value/mean value/highest value on that metric among all the ontologies.
[b] Clarity: only includes words with 1 sense (0.0 represents extreme ambiguity, 1.0 represents no ambiguity).

The metric suite also has implications for developers of ontology languages. The DAML ontologies were developed specifically for the Semantic Web. Most DAML ontologies provide classes, subclasses, and properties in a domain. More detailed information is rarely provided. Information on cardinalities, for example, is only found in approximately one third of the ontologies. More work is needed to determine why ontology designers are not using more advanced features. These features may not be necessary or developers may not understand them.

There is also a need for alternative evaluation methodologies. The methodology operates primarily at a domain or generic level. The evaluation could be extended by including a higher-level (formal) ontological evaluation to identify ontological inconsistencies [19]. Alternatively, the evaluation could be extended to include a more application-specific evaluation of the quality of the elements of the knowledge in an ontology for use in a specific task (e.g., a real-time query). Learning mechanisms would be useful to update evaluations based upon feedback from agents. More work is also needed on the weighting scheme. This initial validation of the metrics used a simple additive weighting scheme. Additional testing and evaluation is needed to derive weights empirically. Empirical testing is also needed across other ontology libraries to determine if our results for the DAML library are generalizable. Finally, empirical testing is needed to validate the relationship between an ontology's internal attributes reflected in its metrics and its external attributes such as its usefulness for supporting an application such as the Semantic Retrieval System.

6. Conclusion

A metric suite for ontology auditing has been proposed and a prototype auditor developed to evaluate the effectiveness of ontologies for the Semantic Web. The prototype auditor has been applied to assess the usefulness of the ontologies found in the DAML ontology library. The results revealed that there are a number of areas where developers need to improve the quality of their ontologies. The research also highlighted the need for future work on ontology evaluation.

Acknowledgment

The authors wish to thank Xinlin Tang and Sunyoung Cho for comments on a previous draft and the editors, reviewers, and attendees of the *Eighth International Conference on Applications of Natural Language to Information Systems*. This research was supported by Georgia State University and by Oakland University.

References

[1] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Scientific American, May 2001, 1–19.

[2] M. Bunge, Treatise on basic philosophyOntology 1: The furniture of the world, vol. 3, Reidel, Boston, 1977.

[3] A. Burton-Jones, S. Purao, V.C. Storey, Context-aware query processing on the semantic web, in: Proceedings of the 23rd International Conference on Information Systems, Barcelona, Spain, 16–19 December 2002.

[4] A. Burton-Jones, V.C. Storey, V. Sugumaran, S. Purao, A heuristic-based methodology for semantic augmentation of user queries on the web, in: Proceedings of 22nd International Conference on Conceptual Modeling, Chicago, Illinois, 2003.

[5] CACM, Special issue on ontology, Communications of the ACM, 45, 2, February 2002, pp. 39–65.

[6] S.R. Chidamber, C.F. Kemerer, A metrics suite for object-oriented design, IEEE Transactions on Software Engineering 20 (6) (1994) 476–493.

[7] O. Corcho, M. Fernandez-Lopez, A. Gomez-Perez, Methodologies tools and languages for building ontologies: where is their meeting point? Data & Knowledge Engineering 46 (2003) 41–64.

[8] O. Corcho, A. Gomez Perez, Evaluating knowledge representation and reasoning capabilities of ontology specification languages, in: Proceedings of ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin, 2000.

[9] Y. Ding, D. Fensel, M. Klein, B. Omelayenko, The semantic web: yet another hip? Data & Knowledge Engineering 41 (2002) 205–227.

[10] R.G. Dromney, Cornering the chimera, IEEE Software 13 (1) (1996) 33–43.

[11] C. Fellbaum (Ed.), Wordnet: An Electronic Lexical Database, MIT Press, Cambridge, MA, 1998.

[12] D. Fensel, F.V. Harmelen, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, Oil: an ontology infrastructure for the semantic web, IEEE Intelligent Systems (2001) 38–45.

[13] M.S. Fox, M. Burbuceanu, M. Gruninger, An organization ontology for enterprise modelling: preliminary concepts for linking structure and behavior, Computers in Industry 29 (1996) 123–134.

[14] D. Gefen, D.W. Straub, M.C. Boudreau, Structural equation modeling and regression: guidelines for research practice, Communications of the AIS 4 (7) (2000) 1–77.

[15] A. Gelbukh, G. Sidorov, Algorithm of word sense disambiguation in an explanatory dictionary, in: Proceedings of COMPLEX-2001, Workshop on Computational Lexicography, Great Britain, 2001.

[16] A. Gelbukh, G. Sidorov, Automatic selection of defining vocabulary in an explanatory dictionary, in: Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 2276, Springer-Verlag, 2002, pp. 300–303.

[17] J. Greenberg, Automatic query expansion via lexical–semantic relationships, Journal of the American Society for Information Science 52 (5) (2001) 402–415.

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

[19] N. Guarino, C. Welty, Evaluating ontological decisions with Ontoclean, Communications of the ACM 45 (2) (2002) 61–65.

[20] R.V. Guha, D.B. Lenat, Enabling agents to work together, Communications of the ACM 37 (7) (1994) 127–142.

[21] J. Heflin, J. Hendler, A portrait of the semantic web, IEEE Intelligent Systems (2001) 54–59.

[22] J. Hendler, Agents and the semantic web, IEEE Intelligent Systems (2001) 30–36.

[23] IEEE, Special issue on the semantic web, IEEE Intelligent Systems (2001) 32–79.

[24] N. Lammari, E. Metais, Building and maintaining ontologies: a set of algorithms, Data & Knowledge Engineering 48 (2003) 155–176.

[25] Y. Ledo-Mezquita, G. Sidorov, A. Gelbukh, Tool for computer-aided Spanish word sense disambiguation, in: Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 2588, Springer-Verlag, 2003, pp. 279–282.

[26] A. Maedche, S. Staab, Ontology learning for the semantic web, IEEE Intelligent Systems (2001) 72–79.

[27] G.A. Miller, Contextuality, in: J. Oakhill, A. Garnham (Eds.), Mental Models in Cognitive Science, Psychology Press, East Sussex, UK, 1996, pp. 1–18.

[28] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K.J. Miller, Introduction to wordnet: an on-line lexical database, International Journal of Lexicography 3 (4) (1990) 235–244.

[29] B. Smith, Ontology and information systems, Stanford encyclopedia of philosophy, forthcoming (down loaded from <http://ontology.buffalo.edu/ontology(PIC).pdf>, September, 23, 2003).

[30] R. Stamper, K. Liu, M. Hafkamp, Y. Ades, Understanding the role of signs and norms in organizations—a semiotic approach to information systems design, Behaviour & Information Technology 19 (1) (2000) 15–27.

[31] L.M. Stephens, M.N. Huhns, Consensus ontologies: reconciling the semantics of web pages and agents, IEEE Internet Computing (2001) 92–95.

[32] V. Sugumaran, V.C. Storey, Ontologies for conceptual modeling: their creation use and management, Data & Knowledge Engineering 42 (3) (2002) 251–271.

[33] Y. Wand, R. Weber, On the deep structure of information systems, Journal of Information Systems 5 (1995) 203–223.

[34] R. Weber, Ontological issues in accounting information systems, in: S. Sutton, V. Arnold (Eds.), Researching Accounting as an Information Systems Discipline, American Accounting Association, Sarasota, FL, 2002.

Andrew Burton-Jones is a doctoral candidate in Computer Information Systems at Georgia State University. He holds a Bachelor of Commerce with First Class Honors and a Masters in Information Systems from the University of Queensland, Australia. He has two research streams. The first stream seeks to obtain a deeper understanding of the meaning and consequences of system usage in organizations. The second stream seeks to improve methods for analyzing and designing systems so that they meet user requirements. He has published in the *Database for Advances in Information Systems* and several international conferences. His awards include the Best Research Paper Award at the International Conference on Information Systems, 2002.

Veda C. Storey, Professor of Computer Information Systems and Computer Science, Georgia State University, has research interests in ontologies, intelligent systems and real world knowledge. Her research has been published in *ACM Transactions on Database Systems*, *IEEE Transactions on Knowledge and Data Engineering*, *Information Systems Research*, *Management Information Systems Quarterly*, *Data and Knowledge Engineering*, *Decision Support Systems*, and *Information & Management*. She has served on the editorial board of a number of journals including *Management Science*, *Information Systems Research*, *Management Information Systems Quarterly*, *Data Base*, *Decision Support Systems*, *Data and Knowledge Engineering*, and the *Journal of Data Management*. Dr. Storey is the program co-chair for the *International Conference on Conceptual Modeling* (*ER 2000*) and for the *International Conference on Information Systems* (*ICIS 2001*). Dr. Storey received her doctorate in Management Information Systems from the University of British Columbia, Canada. She earned a Master of Business Administration degree from Queen's University, Ontario, Canada, and a Bachelor of Science degree from Mt. Allison University, New Brunswick, Canada. In addition, she received her Associate of the Royal Conservatory of Music for flute performance from the University of Toronto, Canada.

Vijayan Sugumaran is an Associate Professor of Management Information Systems in the department of Decision and Information Sciences at Oakland University, Rochester, Michigan, USA. His research interests are in the areas of ontologies and semantic web, intelligent agent and multi-agent systems, component based software development, knowledge-based systems, and data & information modeling. His most recent publications have appeared in *Communications of the ACM*, *Decision Support Systems*, *Healthcare Management Science*, *Data and Knowledge Engineering*, *The DATABASE for Advances in Information Systems*, *Information Systems Journal*, *Journal of Information Systems and E-Business Management Expert Systems with Applications*, and *Logistics Information Management*. Dr. Sugumaran is the *editor-in-chief* of the *International Journal of Intelligent Information Technologies* and also serves on the editorial board of *Journal of Database Management*, *Journal of Electronic Commerce in Organizations*, *Journal of Computer Information Systems*, *Industrial Management and Data Systems*, *International Journal of Information Technology and Web Engineering*, and *Journal of International Technology and Information Management*. He is the Chair of *Intelligent Information Systems* track for the Information Resources Management Association International Conference (IRMA 2001, 2002, 2005) and the *Intelligent Agent and Multi-Agent Systems in Business* mini-track for Americas Conference on Information Systems (AMCIS 1999–2005).

Punit Ahluwalia is a doctoral candidate in Computer Information Systems at Georgia State University. He holds a Bachelor of Engineering (Electrical) from Regional Engineering College, Kurukshetra, India and a Masters in Information Systems from Georgia State University, Atlanta, Georgia, USA. His research interests are quality of service in wireless networks, perceived quality and satisfaction in information systems, and the role of users' requirements in designing and operating information systems. He has published in the *International Journal for Mobile Communications* and in several international conferences.