The Knowledge Engineering Review, Vol. 29:2, 201–216.
doi:10.1017/S0269888914000071

A model to predict quality of a reduced ontology for Web service discovery on mobile devices

DAN SCHRIMPSHER and LETHA ETZKORN

Computer Science Department, University of Alabama in Huntsville, Huntsville, AL, USA;
e-mail: dschrimpsher@acm.org, letzkorn@cs.uah.edu

Abstract

As Web Services and the Semantic Web become more important, enabling technologies such as Web service ontologies will grow larger. At the same time, use of mobile devices to access Web services has doubled in the last year. The ability of these resource-constrained devices to download and reason across ontologies to support service discovery are severely limited. Since concrete agents typically only needs a subset of what is described in a Web service ontology to complete their task, a reduced ontology can be created. Measuring the quality of a reduced ontology, in both knowledge content and performance, is a nontrivial task. Expert analysis of the ontologies is time-consuming and unreliable. We propose two measures of knowledge content and performance. Mean average recall (MAR) with respect to the original ontology compares the data returned from a series of queries related to a particular concept of interest. Mean average performance (MAP) compares the download and reasoning speedup of the reduced ontology with respect to the original ontology. Neither of these values can be computed easily, therefore we propose a set of ontology metrics to predict these values. In this paper, we develop two prediction models for MAR and MAP based on these metrics. These models are based on analysis of 23 ontologies from five domains. To compute MAR, a specific set of queries for each domain was applied to each candidate reduced ontology along with the original ontology. To compute MAP, a simulated mobile device will download and process of each ontology along with the original ontology. We believe this model allows a speedy selection of a reduced ontology that contains the knowledge content and performance speedup needed by a mobile device for service discovery.

1 Introduction

Web services are becoming more important to the daily use of computers. People use Web services to find the weather, travel schedules, and the best restaurant while traveling. As tasks get more complex, allowing the computer to do more of the work is necessary. The Semantic Web addresses this concern. Ontologies allow Web services to be machine readable, and thus software agents can handle more of the decision making in completing a task. Functioning as a Web Service API, they allow users to use services that were independently developed to work a task, facilitate the use of agents to collect, process, and exchange information necessary to complete the task, and permit Web service requests based on a concept.

While the use of ontologies to support the Semantic Web is growing, so is the use of resource-constrained computers, such as mobile devices. The International Data Corporation predicts that mobile devices shipments are expected to reach 82 million by 2011 as reported by Cellular-News (2007). With enhanced Web browsing and processing ability, mobile devices are becoming more like computers and less like simple PDAs and phones. A study of mobile device use found that the users accessing Web services doubled in 2009 to over 63 million with 22 million doing so daily as reported in Mobile Marketer (2009).

However, while these machines are evolving in capability, they are limited by power usage and therefore performance constrained for the foreseeable future. At the same time, ontologies and Web services are becoming larger and more complex. The Full Suggested Upper Merged Ontology (SUMO) is a proposed high level ontology that other application specific ontologies could build upon. At present it contains 20 000 terms, 70 000 axioms, and 3000 rules from Ontology Portal (2009). Seidenberg and Rector (2006) found, using the GALEN medical ontology, that most ontology reasoners currently available cannot handle the full ontology and it must be segmented to be processed on a personal computer. Clearly, it would be difficult for a resource-constrained device to download and process an ontology this large in a reasonable amount of time.

It is impractical to expect a Blackberry to download and reason across an ontology with 10 000 nodes in a reasonable time, when it only needs a small portion of the ontology to perform its work. As Manning *et al*. (2006) observed, there is a computational penalty if an agent has to perform reasoning across a large ontology, if it only requires a small subset of it. In these cases, a reduction of the service ontology, geared to the agent's request, would enable the agent to access what is necessary to complete its task.

Creating a reduced ontology of high quality is a nontrivial task. Such a reduced ontology must be complete enough to allow usefulness, but contain minimal extraneous information. It must also be small enough for the agent to handle. If we are seeking more automation in Web service discovery and use, it makes sense to automate the reduction process. The question is, when is a reduced ontology 'good enough' to be used instead of the original complete ontology?

We first considered the idea of using the Web server containing an ontology to perform the reasoning and return the results to the client device. This is the typical model when using ontologies as a knowledge store or for concept translation. When using ontologies for service discovery, however, there are a few drawbacks to this approach.

First, it limits the flexibility of the client in making decisions based on what is contained within the ontology. Reasoning might be adapted as the structure and concepts of the ontology are discovered. It would be difficult, both conceptually and with respect to bandwidth, to pass enough information to the server in advance to handle the user's full task.

Second, it prevents the agent from caching the ontology locally and reusing it for a later request. The ontology structure would be hidden from the client if the server simply returned the results. Any change in the client's request would require communication between the agent and the server and would use more of the mobile device's limited bandwidth.

Finally, forcing the service to provide reasoning for multiple requests simultaneously impacts the server's resources. For a gateway Web server that takes thousand of requests a minute, this could lead to a time delay for each request. Using our reduction approach, the server would take a performance hit in creating the reduced ontologies, but the download time, reasoning time, and memory requirements for the mobile device would be significantly shortened, giving an overall performance increase. With limited resources available, optimizing the mobile device performance is more important.

In this paper, our aim is to establish a predictive model that uses metrics to evaluate the quality and performance of a reduced ontology. The quality of a reduced ontology can be defined in two parts. First, it needs to be small relative to the original ontology, to make the sub-graphing worthwhile. Second, it needs to retain all of the information in the ontology relative to the concept of interest (COI). We measure these two aspects of quality using an information measure and a performance measure defined in Section 3.

Our approach is dynamic, meaning it can operate at runtime, as clients make requests. It employs easy-to-calculate metrics combined in a predictive model based on previous statistical evidence, allows a dynamic creation of reduced ontologies, as well as speedy selection of reduced ontologies that will be good substitutes for the original ontology. By using these smaller reduced ontologies for downloading and reasoning over, the overall performance of the mobile devices are improved.

A linear regression model was created for information content in a previous study (Schrimpsher & Etzkorn, 2009b). However, this study only focused on a single domain, travel. In order to have confidence in a domain-independent model, as well as creating a model for performance, this study is looking at multiple domains of ontologies in order to create the model.

In this paper, we create a linear regression model of previously defined ontology metrics (Yao *et al.*, 2005; Orme *et al.*, 2006a, 2006b; Schrimpsher & Etzkorn, 2009a) to predict both the informational quality and performance of the reduced ontologies. Section 2 gives an overview of ontologies, ontology metrics, and previous work to reduce ontologies. Section 3 provides the methods we used to conduct this study. Section 4 provides the statistical results and models created. Finally we will conclude and discuss future work in Section 5.

2 Background

2.1 Web service ontology

Vrandecic and Sure (2007) define a Web service ontology as both the terminology and information required to access Web services by providing a machine-readable description of a group of services and their consequences. Martin *et al.* (2004) demonstrated this ability through a specific implementations of OWL, known as OWL-S, which was designed to work with existing Web service protocols, including the Web service description language (WSDL) and the Simple Object Access Protocol (SOAP). OWL-S enhances WSDL with the ability to incorporate semantics into Web service discovery and use. The utility of OWL-S has been demonstrated through its use in enhancing Amazon's WSDL-based Web services. In this project, it was demonstrated that through ontology-based semantics a client agent can query Amazon's database without requiring human interaction, as would have been necessary if using only WSDL.

Hull (2005) advanced the Semantic Web Service Ontology initiative given by DAML (2005) by integrating the Process Specification Language with Web services. He integrated OWL-S style atomic services, WSDL like messages, and process and data flow models of the Web services. This yields semantically accessible Web service discover and translations between semantic concepts on the client and server sides.

These are some examples of how semantic Web services are incorporating ontologies into their methods for discovery and communications. Our research is not tied to these technologies, however, and is flexible enough to work with emerging types of semantic Web services that incorporate ontologies.

2.2 A reduced ontology

Seidenberg and Rector (2006) demonstrated that very large ontologies are difficult to process and reason over. In the database world, large data sets are filtered down into smaller sets through views. There has been some research into creating a database type 'view' for ontologies (e.g. Stuckenschmidt & Klein, 2003; Volz *et al.*, 2003; Noy & Musen, 2004; Jiménez-Ruiz *et al.*, 2007a, 2007b). This idea treats the full ontology as a database and queries for one or more COIs.

To help solve this problem, Jiménez-Ruiz *et al.* (2007b) built a Protege-based query language, called OntoPath. Much like a database query, OntoPath allows users to query an ontology for concepts and relationships to retrieve an ontology view, or 'personalized modules'. An OWL ontology is stored in a database where OntoPath can query it much like a traditional database. While OntoPath's XPath-based query language is relatively easy to use, it requires the user to understand more about the ontology being queried than a typical semantic Web user or agent would understand.

Volz *et al.* (2003) acknowledge that the semantic Web must support the 'needs of specific user communities' and create particular information for them. To fill this need, they created a 'view language'. This view language is designed to show users a view of an ontology based on a query or set of queries in an extension of the RDF Query Language (RQL). The views they create are designed to be indistinguishable from the source ontology by the agent. However, they do require the use of an ontology to describe their view language to a user. Updates of these views are also limited and the semantic characterizations of the views cannot be computed automatically. While this is a step in the direction of dynamic reduced ontologies, it still requires the user to build specific queries to get the view they want and does not say anything about the quality of the view created. The burden is placed on the user agent.

In order to limit the information, a user has to process when interacting with large ontologies, Noy and Musen (2004) created a traversal algorithm. This algorithm builds a 'traversal view' from a COI or set of COIs by following property resources out a set distance defined by the user. They define a view built by a query, such as the Jiménez-Ruiz *et al.* (2007b) OntoPath method as an ontology view. However, they realize that this view does not allow a user to see all concepts closely related to their COI. They propose a traversal view to complement the query views to serve this need. Again, although this is a step in the direction of dynamic sub-graphing to create reduced size ontologies, it is still targeted toward users interacting with a tool to access information contained within a classifying ontology, such as a genome.

There has also been research into creating modular rather than monolithic ontologies that could be merged by the agent when they are needed. Stuckenschmidt and Klein (2003) defined an architecture to create modular ontologies during the design and development phase. Their goal is to create modular ontologies with loose coupling, self-containment, and integrity. The architecture uses queries between modules to map the modular ontologies together when needed. This allows reasoning across multiple modules through these mappings. A necessary piece of this architecture is a method for detecting changes in modules that effect a mapping. However, these modules are defined at development time and do not address the specific needs of an agent dynamically.

Kusnierczyk (2008) looked at partitioning the Gene Ontology by using the Taxonomy of Species to define how to automatically generate the slimmed modules. The framework relates concepts in the Gene Ontology to the Taxonomy of Species through validity, specificity, and relevance.

Grau *et al.* (2007) developed a method to extract reusable fragments, or modules, from ontologies based on a given set of terms. They show that minimal module extraction is not algorithmically solvable. Their syntactic locality algorithm was compared with Noy and Musen's (2003) PROMPT-FACTOR algorithm and Grau *et al.*'s (2006) Modularization Algorithm. Grau *et al.* (2007) found in every case, their locality algorithm created smaller modules than either of the other two algorithms using a series of complex and simple ontologies. This work is similar to ours in that Grau *et al.* (2007) wanted to create fragments of large ontologies. It is, however, still a static model designed to create reusable modules rather than a dynamic method, such as our method, for creating reduced ontologies. However, their success with syntactic locality gives confidence that a dynamic locality-based sub-graphing algorithm such as the algorithm we use in our method is a reasonable approach.

2.3 Ontology metrics

Metrics are widely used in software engineering and information retrieval to judge quality. Various metrics to measure cohesion, coupling, and complexity in software engineering, and the use of recall, and precision metrics to determine the accuracy of information retrieval queries and approaches are all familiar topics in these fields (e.g. Etzkorn *et al.*, 2004; Manning *et al.*, 2006). Metrics have recently been applied to ontologies as well. A summary of ontology metrics is given in Table 1.

Much early work in ontology metrics translated standard object-oriented software metrics to apply to ontologies rather than to object-oriented software. For example, Yao *et al.* (2005) applied cohesion metrics to RDF-based ontologies. They defined Number of Roots (NoR), Number of Leafs (NoL), and Average Depth of Inheritance Tree–Leaf Nodes (ADIT-LN) metrics. Then these

Table 1 Overview of ontologies metric studies

Metric	Type	Author	Purpose
Cohesion metrics	Structural	Yao *et al.* (2005)	To measure cohesion within an ontology
Coupling metrics	Structural	Orme *et al.* (2006a)	To measure coupling between ontologies
Complexity metrics	Structural	Orme *et al.* (2006b)	To measure complexity within an ontology
AKTiveRank	Structural	Alani and Brewster (2006)	Ranking ontologies for potential reuse
Stable metrics	Semantic	Vrandecic and Sure (2007)	Dynamic semantic metrics
Analytic metaphysical	Semantic	Guarino and Welty (2004)	Domain independent measures
Hunter's scoring	Semantic	Qi and Hunter (2007)	Measuring ontologies that change over time
Shapely value	Semantic	Deng *et al.* (2007)	Measuring inconsistency in ontologies
Qood grid	Structural Functional Usability	Gangemi *et al.* (2006)	Overall quality of an ontology
Ontometric	Structural Semantic	Lozano-Tello and Gomez-Perez (2004)	Help developers choose the right ontology
OntoQA	Structural Semantic	Tartir *et al.* (2005)	Ontology quality
Reduced metrics	Structural	Schrimpsher and Etzkorn (2009a)	Measure the quality of a reduced ontology

ontologies were evaluated by experts and correlated with the metric values. There was a medium to high correlation between the cohesion metrics and the experts' opinions of the cohesion of the ontology.

In later work, Orme *et al.* (2006a) applied coupling metrics to ontologies. These metrics included number of external classes (NOEC), reference to external classes (REC), and referenced includes (RI). Again they found that these metrics were highly correlated with expert analysis of the coupling of the ontology.

Most recently, Orme *et al.* (2006b) applied complexity metrics to ontologies. These metrics included number of classes (NoC), number of Fan-outs (NoF) and ADIT-LN. Once again these metrics were highly correlated with expert analyses of the ontologies. We can take from these studies a confidence that the quality of ontologies in general can be measured by metrics.

Alani and Brewster (2006) created AKTiveRank, a system for ranking ontologies for their reuse potential. AKTiveRank uses an idea similar to the page rank system of Google. Using the Java Universal Network/Graph framework (JUNG) query system to query the RDF ontologies, AKTiveRank makes four measurements. The first of these metrics is the Class Match Measure (CMM), which measures the coverage of an ontology over a set of terms. Second, the Density measure calculates the information content of a class within an ontology. This includes ideas, such as number of subclasses and number of properties. Third, the Semantic Similarity Measure (SMM) calculates, within an ontology, how close together classes are that match the query. Finally, the Betweenness measure calculates the number of shortest paths that pass through each node in the graph. Alani and Brewster compared their results with expert analyses and found that the CMM matched up best with the experts, but using a weighting of 0.4(CMM), 0.3(BEM), 0.2(SMM), and 0.1(DEM) on the four metrics yielded a total score that was highly correlated with expert opinion.

Vrandecic and Sure (2007) extend ontology metrics from purely structural metrics into semantic metrics. They also introduce the concept of normalizing ontologies prior to computing the metrics, which they argue allows them to 'apply known structural metrics in a semantics-aware way'. A 'stable metric' refers to a metric that maintains its meaning even as new axioms are added to the ontology. They argue that both of these properties allow them to measure semantic metrics in a dynamic environment.

In the same vein, Guarino and Welty seek to find ‘analytic metaphysical’ (Guarino & Welty, 2004) measures of ontologies that are valid regardless of the ontology domain or structure. These include ideas of essence, rigidity, identity, and unity. However, this framework deals with individual properties and not with the quality of the ontology of a whole, except perhaps by uniting the metrics for individual axioms (Guarino & Welty, 2004).

Lozano-Tello and Gomez-Perez (2004) created a tool, ONTOMETRIC, to measure the quality and usefulness of an ontology relative to the system needs. This method uses 160 characteristics, which Lozano-Tello and Gomez-Perez call their multilevel framework of characteristics. Their characteristics were validated by 10 ontology experts. They further categorized these characteristics into five domains including content, language, methodology, tools, and costs. Rather than confine themselves to one ontology language, they build a multilevel tree of characteristics (MTC) based on a reference ontology, which includes generic items, such as ontology, class, attribute, instance, relation, and axiom. ONTOMETRIC is a semi-automated system that works as a decision aid to help developers choose the right ontology given the requirements for the system (Lozano-Tello & Gomez-Perez, 2004).

Tartir *et al.* also created a tool for measuring the quality of an ontology, called OntoQA. This tool analyzes the ontology schema and the knowledge bases it describes through a set of metrics. These metrics are defined in two categorize, schema metrics and instance metrics. Schema metrics include relationship richness, attribute richness and inheritance richness. Instance metrics include two more categories, knowledge base metrics and class metrics. Class richness, average population and cohesion are included in knowledge base metrics, while importance, fullness, inheritance richness, relationship richness, connectivity, and readability make up the class metrics. They use these OntoQA and these metrics to analyze a number of ontologies (Tartir *et al.*, 2005).

We previously (Schrimpsher & Etzkorn, 2009a) defined five new metrics that are intended to measure the information content and performance speedup in reduced ontology relative to the original ontology. The informational metrics include Proximity, Relevance, and Completeness (Schrimpsher & Etzkorn, 2009b). The performance metrics include Size and Compactness. These metrics were measured, given a specific COI, against mechanically generated reduced ontology candidates. Multiple queries were applied to both the original ontology and the candidate ontologies to compute mean average recall (MAR). A significant correlation was found between all the informational metrics and MAR (Schrimpsher & Etzkorn, 2009b). A linear regression model using these metrics along with other structural metrics was constructed for a group of travel agency ontologies (Schrimpsher & Etzkorn, 2009b). However, by building the model on a single domain, it runs the risk of being domain specific. In this new study, we expand the model to multiple domains to create more confidence it is domain agnostic and can operate over any domain.

3 Research description

The purpose of this study is to demonstrate that some subset of the structural metrics chosen can predict both the information quality of a reduced ontology compared with the original ontology, and the performance advantage provided by the reduced ontology relative to the original ontology. First, we choose the ontologies and queries for the study. Then reduced ontology candidates were generated automatically for each ontology. Next, MAR and MAP were computed for each reduced ontology relative to its original ontology. Finally, the selected metrics were applied to each reduced ontology candidate and used to generate a regression model to predict MAR and MAP. Each of these steps is described in detail in the following sections.

3.1 Definition of quality and performance

We define a ‘good’ reduction in two ways, informational quality and performance. When creating a reduced ontology, the independent quality of the base ontology is ignored. Rather, the base

Table 2 Overview of ontologies selected for study

Domain	Number	Ontologies COI
Animals	4	Birds
Food	4	Fish
Organizational structure	4	Student
Travel	5	Flight

ontology is considered the 'gold standard' from an information retrieval perspective. Information quality determines how well COI queries were answered on the reduced ontology, relative to how those same queries previously were answered on the base ontology. On the other hand, performance measures the download and processing speed improvement for the client for the reduction versus the original ontology.

To measure informational quality, we use MAR. From information retrieval, recall is defined as the ratio of the number of items returned from a query in the reduced ontology to the number of items returned in the original ontology. Recall is computed for each of the queries for an ontology and the mean of these recall scores forms the MAR. Thus, MAR measures the average amount of information retained in a reduced ontology relative to the COI.

To measure performance, we use MAP. Performance is computed by taking the ratio of time required for a mobile device to download and process a reduced ontology versus its base ontology. This measures the speedup of handling the reduced ontology. This is performed in a simulated mobile device environment multiple times. The average speed up is the MAP score.

3.2 Ontology and concept of interest selection

In this study, we selected four domains to pull ontologies from. There are 21 ontologies in total coming from the domains of food (5), travel (4), animals (4), and organizational structure (4). These ontologies were implemented in OWL, with a total of 829 named classes, to analyze in this paper. A common COI was chosen for each domain as a sample request a mobile device's agent might make. These COIs are show in Table 2.

Based on the COI, seven queries for each domain were created by five unaffiliated volunteers. These queries represent information a task a client using a mobile device might to perform in that domain. These queries were applied to each ontology in the domain. The results of these queries are used to calculate MAR for each reduced ontology.

3.3 Constructing reduced ontologies

An OWL ontology can be viewed as a graph, where the nodes are vertices and relationships, including both inheritance and property relations, are edges. A reduced ontology is defined as a subset of nodes and relationships from a base ontology that together form a smaller ontology. A reduced ontology should contain a subset of information contained in the base ontology specific to a COI and exclude non-relevant information.

A modified traversal algorithm created in a previous study (Jiménez-Ruiz *et al.*, 2007b) was run on the ontologies with each node as the center. Each ontology generated N sub-graphs, where N is the number of nodes in the original ontologies. There is no claim this is the only way, or even the best way, to create reduced ontologies, but it is efficient and mechanical. The traversal starts with the selected center node and follows all of its references (property, parents, and children). Each new node is added to the sub-graph and the algorithm is recursively called with that node as the center. This process continues until either there are no more new nodes referenced or the path length passes PLthreshold. Children classes are the same as their parents and thus hold all the parents' references as their own. For that reason the relationship is traversed, but the reference is not counted toward PLthreshold.

Each class defined in each of the ontologies was selected as the center for the traversal algorithm and a calculation was run with a threshold path length of two. Thus, from the 814 nodes in the original ontologies, 814 reduced ontology candidates were created, each one with the center set to a unique node in its original ontology.

3.4 Computing MAR and MAP

Informational quality is a somewhat subjective idea. Previous work has developed various methods for create subsets or views of an ontology. However, the only attempt to validate the ontologies was with Gangemi *et al.*, who matched information retrieval with expert opinions (Gangemi *et al.*, 2005). In order to develop a dynamic model, we need a computable measure of quality. For this study, we have chosen to use MAR to measure informational quality.

Five volunteers unrelated to this research were asked to create typical queries they would have about concepts in the domain. These volunteers had no knowledge of the specific ontologies being used for this study. These volunteers created 28 queries to use in this study. It should be noted that the queries were written in English and had to be mechanically converted to fit the naming conventions of each original ontology. However, no change to the structure or purpose of the query was performed.

The querying of both the original ontology and all the reduced ontology candidates was performed using SPARQL (Stuckenschmidt & Klein, 2003) after loading each ontology into the Jena OWL parser (Etzkorn *et al.*, 2004). The set of queries was applied to all 814 ontologies (the originals and all the reduced ontology candidates).

The recall score for each query was computed and the MAR for the reduced ontology candidates was computed over all the queries. As we discussed earlier, we are only concerned with recall with respect to the original ontology.

Performance though less subjective is still a broad term. There have been some studies into the performance of subsets versus full ontologies (Grau *et al.*, 2007). However, performance has typically been defined as simple size and has not been related to any metrics. We must first define what is meant by performance.

We choose to define performance gain in two parts, bandwidth and processing time. As with quality, the original ontology forms the basis for performance measurement. We measure the bandwidth requirements and processing time of the base ontology and then for the reduced ontology candidates.

In order to make these measurements, a simulation of a mobile device, with various processing capability and bandwidths, was created. This simulation interacts with a mock semantic Web service provider. A series of runs with various capabilities for bandwidth and processing were completed with noise applied to the actual amount of resources the simulated device contained. From the results we computed average bandwidth and processing time required for the original ontology and its reduced ontology candidates. The ratio of average time for the original ontology and each reduced ontology candidate was set to the MAP for that reduced ontology.

3.5 Applying the metrics

While metrics have been in use for software systems for many years, applying them to ontologies is a relatively new idea. There have been metrics previously defined to describe the structure of an ontology (Yao *et al.*, 2005; Orme *et al.*, 2006a, 2006b). We chose the structural metrics because the are computable in a short time, making them useful for a dynamic model. Other, semantic type metrics, are difficult to compute in a short time and therefore not as useful for this model.

None of the metrics defined for general ontology quality give a look at reduced ontologies specifically. So we define five new reduced ontology metrics in a previous study (Schrimpsher & Etzkorn, 2009a) in

Table 3 Ontology metrics used in this study

Metric	Description
NOEC	Number of external classes included in the ontology (Orme *et al*., 2006a).
REC	References to external classes included in the ontology (Orme *et al*., 2006a).
RI	Number of references included in the ontology (Orme *et al*., 2006a).
RC	Number of root classes in the ontology (Yao *et al*., 2005).
LC	Number of leaf classes in the ontology (Yao *et al*., 2005).
ADIT-LN	Average depth of the inheritance tree in the ontology (Yao *et al*., 2005).
NoC	Number of classes in the ontology (Orme *et al*. 2006b).
NoF	Number of fan-outs in the ontology (Orme *et al*., 2006b).
Relevance	Ratio of unreachable nodes to the total nodes (Schrimpsher & Etzkorn, 2009a).
Proximity	Ratio of length to external references vs the original ontology (Schrimpsher & Etzkorn, 2009a).
Completeness	Ratio of external references to the total references (Schrimpsher & Etzkorn, 2009a).
Compactness	Ratio of the path length the maximum path length (Schrimpsher & Etzkorn, 2009a).
Size	Ratio of the number of nodes to the original ontology (Schrimpsher & Etzkorn, 2009a).

NOEC = number of external classes; REC = reference to external classes; RI = referenced includes; NoC = number of classes; NoF = number of fan-outs.

Table 4 Cohen's correlation magnitude scale

Range	Category
0.0–0.1	Trivial
0.1–0.3	Minor
0.3–0.5	Moderate
0.5–0.7	Large
0.7–0.9	Very large
0.9–1.0	Almost perfect

order to look at how well a reduced ontology matches the original ontology. Table 3 represents a description of structural metrics chosen for this study.

The metrics were applied to each reduced ontology candidate and the original ontology. For this study, COIs are limited to a single choice of a node in the ontology. Since Jena provides an indexed list of the nodes in the ontology, checking for a single node is a trivial task.

We then computed the metrics defined in Table 3 on each of the remaining reduced ontologies. Each metric was automatically computed by a tool the authors developed in Java based on the Jena OWL parser. Each of the metrics defined in Table 3 is then correlated with the MAR score. This correlation provides a filter on which metrics are actually useful at predicting quality. This will provide confidence in which metrics are useful to quality and which may not be.

Each of the metrics defined in Table 3 is then correlated with the MAP score. This correlation provides a filter on which metrics are actually useful at predicting performance. This will provide confidence in which metrics are useful to performance and which may not be. We used Cohen's correlation magnitude scale (Cohen, 1998) shown in Table 4.

3.6 Building the model

Linear regression is a well understood way to finding linear models between observed data and responses. However, it does suffer from sensitivity to intercorrelation, so the first step in creating this a linear regression model is finding and removing metrics with intercorrelation. Once the metrics to be used in the regression are decided, a step-wise regression is performed to find out which metrics provided the best model.

Table 5 The correlations between ontology metrics and MAR

Ontology metric	Correlation to MAR	*p*-value	Cohen's magnitude scale
Proximity	0.773	<0.0005	Very large
Size metric	−0.708	<0.0005	Very large
Compactness	0.689	<0.0005	Large
Relevant nodes	0.593	<0.0005	Large
NoC	0.585	<0.0005	Large
NoF	0.536	<0.0005	Large
NOEC	0.520	<0.0005	Large
leafClasses	0.425	<0.0005	Moderate
Completeness	0.411	<0.0005	Moderate
rootClasses	0.408	<0.0005	Moderate
ADITN	0.344	<0.0005	Moderate
REC	0.341	<0.0005	Moderate
RI	0.147	0.057	Minor

NoC = number of classes; NoF = number of fan-outs; NOEC = number of external classes; MAR = mean average recall; REC = reference to external classes; RI = referenced includes.

A preliminary attempt at creating a linear model between metrics and MAR in a single domain was performed previously (Schrimpsher & Etzkorn, 2009b). This study expands those results to include models predicting MAP as well as MAR and using multiple domains. The plan is create two linear regressions (one for MAR and one for MAP) based on the 829 reduced ontology candidates in the four domains chosen. Once the two models are found, a standard analysis will be performed.

4 Results

4.1 MAR and MAP correlations

In the first part of this study, each metric was correlated with the MAR and MAP scores. These correlations were ranked according to Cohen's magnitude scale (Cohen, 1998). The MAR correlations are shown in Table 5. Two metrics have a very large correlation with MAR, proximity and size. Compactness, relevant nodes, NoC, NoF, and NOEC all have large correlations. The rest have moderate correlations except for RI, which has a minor correlation. This is promising for creating a useful model, since all but one of the metrics have at least a moderate correlation with MAR.

The MAP correlations are shown in Table 6. RI is the only metric with a large correlation with MAP. Completeness, ADITN, and NOEC all have moderate correlations. The rest of the metrics have minor correlations. While this is less promising than the MAR results, it is still possible we can create a good model to predict MAP. However, it may be there are variables mission from this model that are needed to estimate MAP.

4.2 Intercorrelation

In order to create a linear regression model, we must first identify and remove any intercorrelated metrics, since these will bias the results and give us an unreliable model. As a threshold, we set a correlation of 0.5 to say two metrics were intercorrelated. Table 7 shows each metrics and which other metrics it is intercorrelated with.

In order to ensure that no intercorrelated metrics are in the models, we created a list of possible variables for the models by choosing all combinations of metrics. We started with the highest correlation to MAR and MAP and then added metrics that were not correlated with any variables already in the model.

Table 6 The correlations between ontology metrics and MAP

Ontology metric	Correlation to MAR	*p*-value	Cohen's magnitude scale
RI	−0.533	<0.0005	Large
Completeness	−0.471	<0.0005	Moderate
ADITN	−0.386	<0.0005	Moderate
NOEC	−0.309	<0.0005	Moderate
Size metric	0.255	<0.0005	Minor
NoC	−0.217	<0.0005	Minor
NoF	−0.192	<0.0005	Minor
Proximity	−0.192	<0.0005	Minor
rootClasses	−0.185	<0.0005	Minor
Relevant nodes	−0.175	<0.0005	Minor
Compactness	−0.165	<0.0005	Minor
leafClasses	−0.141	<0.0005	Minor
REC	−0.123	<0.0005	Minor

MAP = mean average performance; MAR = mean average recall; RI = referenced includes; NOEC = number of external classes; NoC = number of classes; NoF = number of fan-outs; REC = reference to external classes.

Table 7 List of intercorrelated metrics

Metric	Inter-correlation
Compactness	Relevant Nodes, Size, Proximity, NOEC
Relevant Nodes	Compactness, Proximity
Completeness	Size, ADITN, NoC, NoF
Size	Compactness, Completeness, Proximity, NOEC, rootClasses, leafClasses, NoC, NoF
Proximity	Compactness, Relevant Nodes, Size, NoC, NoF
NOEC	Compactness, Size, rootClasses, NoC, NoF
REC	rootClasses, leafClasses, NoF
RI	
rootClasses	Size, NOEC, REC, leafClasses, NoC, NoF
leafClasses	Size, REC, rootClasses, NoC, NoF
ADITN	Completeness
NoC	Size, Proximity, NOEC, rootClasses, leafClasses, NoF
NoF	Completeness, Size, Proximity, REC, rootClasses, leafClasses, NoC

NOEC = number of external classes; NoC = number of classes; NoF = number of fan-outs; REC = reference to external classes; RI = referenced includes.

4.3 MAR Model

For the MAR model, a number of variable options were analyzed using a stepwise regression approach. Montgomery *et al.* (2006) defines a stepwise regression is a way of evaluating a number of regression models automatically by evaluating their partial *F* statistics at each step in order to find the best linear regression model. The stepwise regression was done with the Statistical and Process Management Software for Six Sigma and Quality Improvement–Minitab (2010) statistical software package and used a threshold of 0.15 to enter and remove. Based on the results of this stepwise regression, the variables selected were Proximity, NOEC, leafClasses, Completeness, and RI. The model selected was:

$$\widehat{MAR} = -0.0616 + 0.709 \times Proximity + 0.0748 \times NOEC + 0.00250 \times leafClasses + 0.0650 \times Completeness$$

The results of the *t*-test on the predictors are shown in Table 8. A threshold of 0.10 was chosen. The *p*-values are all <0.005 with the exception of Completeness ($p = 0.098$), but they all fall within

Table 8 *T*-test results whether there is a relationship between the metrics and MAR

Predictor	*t*	*p-value*	VIF
Constant	−2.80	<0.0005	na
Proximity	24.63	<0.0005	1.36
NOEC	9.36	<0.0005	1.43
leafClasses	3.25	<0.0005	1.99
Completeness	1.66	0.098	1.60

NOEC = number of external classes; MAR = mean average recall.

Table 9 R^2 statistics for MAR model

R-type statistic	Value
R^2	64.6%
R^2_{adj}	64.4%
R^2_{pred}	63.91%

MAR = mean average recall.

Table 10 ANOVA of MAR linear regression

Source	DF	SS	MS	*F*	*p*-value
Regression	4	57.884	14.471	368.95	<0.0005
Residual error	809	31.731	0.039		
Total	813	89.614			

MAR = mean average recall.

our chosen threshold. All of the Variance Inflation Factors (VIF) are <2, so intercorrelation does not appear to be a problem. The R^2 values are given in Table 9. These were the highest values of all the variable combinations examined. These values give us a high confidence in future predictions. ANOVA analysis (Montgomery *et al.* 2006) shown in Table 10, demonstrates that there is definitely a relationship between the metrics and MAR.

In order to be more confident in our model, we conducted further analysis on the residuals. The standard residual plots are given in Figure 1. The first assumption, normality of the residual data, is checked through the normality plot in the upper left of Figure 1. While there appears to be some non-normal behavior at 0 in the residuals, they do follow the line fairly well and we accept it. This can also be seen in the histogram plot of the residuals, showing a bell shape of the residuals. The second assumption, randomness, is check through the versus fit plot given in the upper right of in Figure 1. The errors seem to be fairly random and no obvious pattern is seen by the authors. The third assumption, that residuals are independent of order, is shown in the versus order plot in the lower right of Figure 1. The residuals seem to be fairly block shaped and there appears to be no order dependence.

So given this analysis, we initially conclude that the model is a good predictor of MAR, and therefore predicts the quality of the reduced ontology relative to retaining information on the COI.

4.4 MAP model

For the MAP model, a number of variable options were analyzed using a stepwise regression approach. The stepwise regression was done with the Minitab statistical software package and used

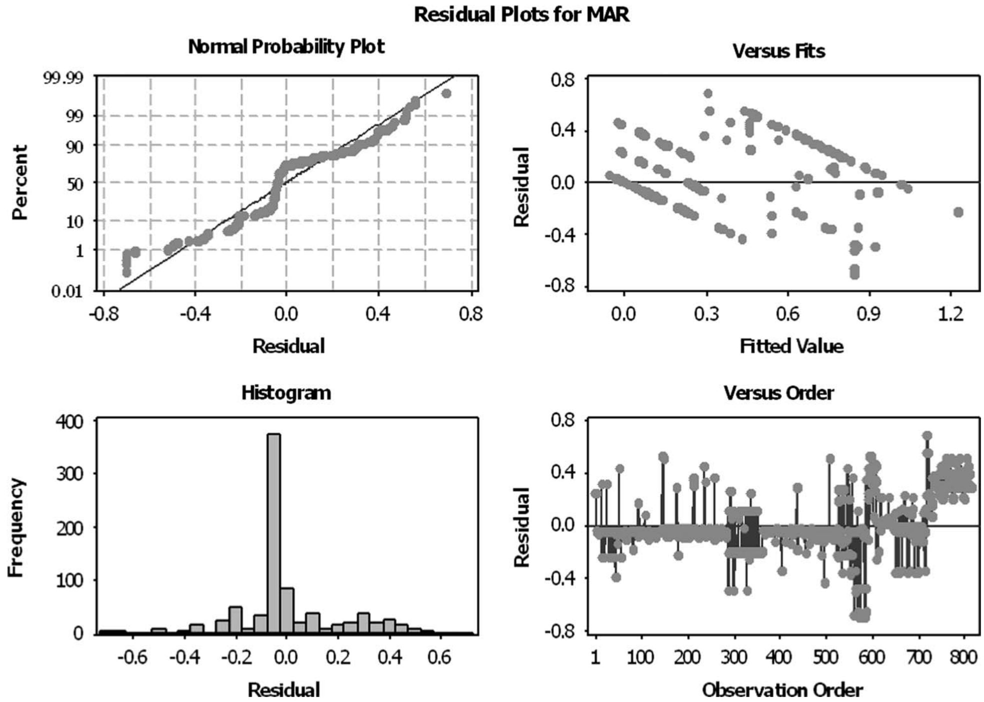

Figure 1 Standard residual plots for MAR. MAR = mean average recall

Table 11 *T*-test results whether there is a relationship between the metrics and MAP

Predictor	t	p-value	VIF
Constant	28.37	<0.0005	na
RI	−25.00	<0.0005	1.12
Completeness	−10.07	<0.0005	1.76
NOEC	−7.39	<0.005	1.40
leafClasses	4.28	<0.005	1.86

MAP = mean average performance; VIF = Variance Inflation Factors; RI = referenced includes; NOEC = number of external classes.

Table 12 R^2 statistics for MAP model

R-Type Statistic	Value
R^2	59.2%
R^2_{adj}	59.0%
R^2_{pred}	55.9%

MAP = mean average performance.

a threshold of 0.15 to enter and remove. Based on the results of this stepwise regression, the variables selected were Proximity, NOEC, leafClasses, Completeness, and RI. The model selected was:

$$\widehat{MAP} = 1458 - 331 \times RI - 108 \times Completeness - 15.3 \times NOEC + 0.831 \times leafClasses$$

The results of the t-test on the predictors are shown in Table 11. An threshold of 0.10 was chosen. The p values are all <0.005 with the exception of Completeness ($p = 0.098$), but they all

Table 13 ANOVA of MAP linear regression

Source	d.f.	SS	MS	F	p-value
Regression	4	3126737	781684	293.41	<0.0005
Residual error	809	2155296	2664		
Total	813	5282034			

MAP = mean average performance.

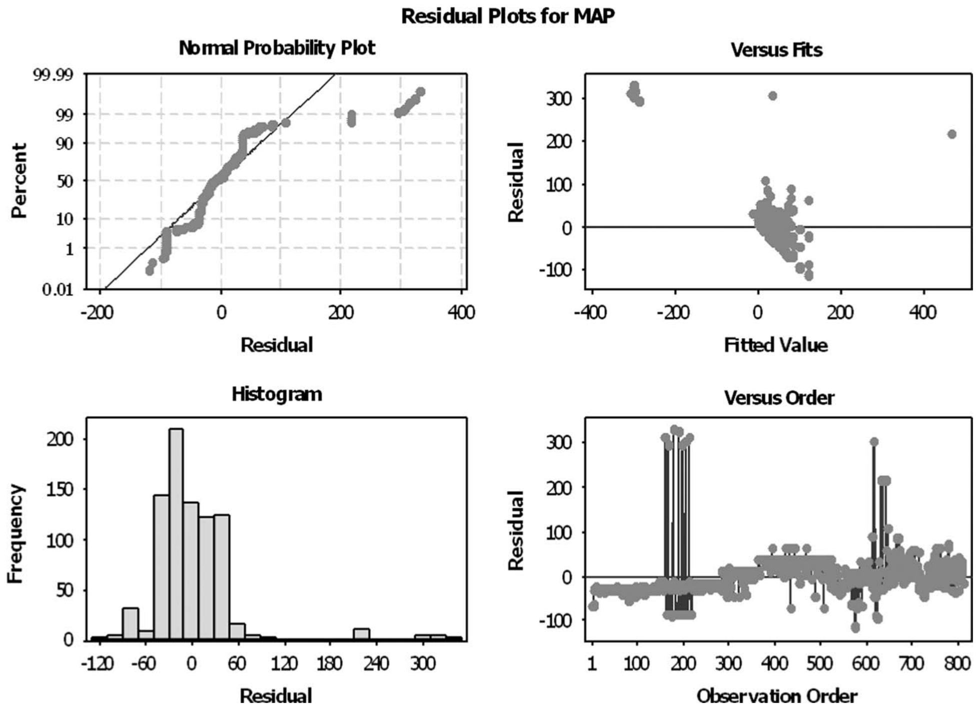

Figure 2 Standard residual plots for MAP. MAP = mean average performance

fall within our chosen threshold. All of the Variance Inflation Factors (VIF) are <2, so intercorrelation does not appear to be a problem. The R^2 values are given in Table 12. These were the highest values of all the variable combinations examined. These values give us a high confidence in future predictions. ANOVA analysis shown in Table 13, demonstrates that there is definitely a relationship between the metrics and MAR.

In order to be more confident in our model, we conducted further analysis on the residuals. The standard residual plots are given in Figure 2. The first assumption, normality of the residual data, is checked through the normality plot in the upper left of Figure 1. There appears to be non-normal behavior for large residuals. However, for lower values of residuals a normal distribution is reasonable. This can also be seen in the histogram plot of the residuals, where the bell shape is skewed to the right. The second assumption, randomness, is checked through the versus fit plot given in the upper right of Figure 1. Again it seems that large residuals make the plot look nonrandom. The third assumption, that residuals are independent of order, is shown in the versus order plot in the lower right of Figure 1. Again very large residuals appear to be outliers, but otherwise the data seems independent of order.

Given this analysis, it is possible that this model is either non-linear, or there is some variable inherent in the model that is not being measured. As this is our first attempt at an MAP model, we are confident that the model can be adjusted, with either a higher order term or a new metric currently not being measured.

5 Conclusions

Creating a true Semantic Web requires handling all types of clients that may use available services. As the size of the Web grows, the size of Web service ontologies will grow as well, out pacing mobile devices' ability to use them. In this research, we have created a workable model to reduce the size of ontologies a mobile device must process while maintaining the core of the information the client needs to perform his task.

Given the results in Section 4, we believe we can now quickly compute informational quality in the form of MAR using a combination of structural ontology metrics. The MAP is not as convincing and requires additional research. That being said, we are now confident an algorithm can be developed to generate high-quality reduced ontologies dynamically.

The next step in this research is to validate these models against ontologies not used in the constructing the models. This will provide more confidence in the models to move forward. As this research continues, the model created will be used to dynamically create reduced ontologies at the time of the user requests. This would allow a Web service ontology to be tailored to a specific task a software agent is engaged in.

Acknowledgment

The authors would like to thank Sampson Gholston for his statistical expertise and Anthony Orme for his out of the box thinking.

References

Alani, H. & Brewster, C. 2006. Metrics for ranking ontologies. In *Proceedings of the 4th International EON Workshop, 15th International World Wide Web Conference (WWW2006)*, Edinburgh, UK.

Cellular-News. 2007. *Converged Mobile Devices Adoption to Reach 82 Million Units by 2011*. http://www.cellular-news.com/story/24073.php

Cohen, J. 1998. *Statistical Power Analysis for the Behavioral Sciences*, 2nd edn. Lawrence Erlbaum Publishing Company.

Deng, X., Haarslev, V., Shiri, N., Franconi, E., Kifer, M. & May, W. (eds) 2007. Measuring inconsistencies in ontologies. *Lecture Notes in Computer Science* **4519**, 326–340.

Etzkorn, L., Gholston, S., Fortune, J., Stein, C., Utley, D., Farrington, P. & Cox, G. 2004. A comparison of cohesion metrics for object-oriented systems. *Information and Software Technology* **46**, 667–687.

Gangemi, A., Catenacci, C., Massimiliano, C. & Lehmann, J. 2005. A theoretical framework for ontology evaluation and validation. In *Proceedings of Semantic Web Applications and Perspectives (SWAP2005)*, Trento, Italy.

Gangemi, A., Catenaccia, C., Ciaramita, M. & Lehmann, J. 2006. Qood grid: A metaontology based framework for ontology evaluation and selection. In *Proceedings of the 4th International EON Workshop, 15th International World Wide Web Conference (WWW2006)*, Edinburgh, UK.

Grau, B., Kazakov, Y. & Sattler, U. 2007. Just the right amount: extracting modules from ontologies. In *Proceedings of the 16th International Conference on World Wide Web (WWW2007)*, Banaff, Canada.

Grau, B., Parsia, B., Sirin, E. & Kalyanpur, A. 2006. Modularity and Web Ontologies. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR-06)*, Lake District of the United Kingdom.

Guarino, N. & Welty, C. 2004. Evaluating Ontological Decisions with ONTOClean. *Communications of the ACM* **45**(2), 61–65.

Hull, R. 2005. Web services composition: a story of models, automata, and logics. In *Proceedings of the IEEE International Conference on Services Computing (SCC2005)*, Orlando, FL.

Jiménez-Ruiz, E., Berlanga, R., Nebot, V. & Sanz, I. 2007a. OntoPath: a language for retrieving ontology fragments. *Lecture Notes in Computer Science* **1**, 897–914.

Jiménez-Ruiz, E., Nebot, V., Berlanga, R., Sanz, I. & Rios, A. 2007b. A protege plug-in-base system to manage and query large domain ontologies. In *Proceedings of 10th International Protégé Conference*, Budapest, Hungary.

Kusnierczyk, W. 2008. Taxonomy-based partitioning of the Gene Ontology. *Journal of Biomedical Informatics* **41**, 282–292.

Lozano-Tello, A. & Gomez-Perez, A. 2004. OntoMetric: a method to choose the appropriate ontology. *Journal of Database Management* **15**, 1–18.

Manning, C., Raghavan, P. & Schütze, H. 2006. *An Introduction to Information Retrieval*. Cambridge University Press.

Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N. & Sycara, K. 2004. Bringing semantics to Web services: the OWL-S approach. In *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, San Diego, CA.

Mobile Marketer. 2009. *Daily Mobile Web Consumption of News, Information Doubles: comScore*. http://www.mobilemarketer.com/cms/news/research/2842.html

Montgomery, D., Peck, E. & Vining, G. 2006. *Introduction to Linear Regression Analysis*. Wiley-Interscience.

Noy, N. & Musen, M. 2003. The PROMPT suite: interactive tools for ontology mapping and merging. *International Journal of Human-Computer Studies* **6**(59), 983–1024.

Noy, N. & Musen, M. 2004. Specifying ontology views by traversal. In *Proceedings of the Third International Semantic Web Conference (ISWC2004)* **3298**, 713–725.

Ontology Portal. 2009. *Suggested Upper Merged Ontology (SUMO)*. http://www.ontologyportal.org/

Orme, A., Yao, H. & Etzkorn, L. 2006a. Coupling metrics for ontology-based systems. *IEEE Software* **23**(2), 102–108.

Orme, A., Yao, H. & Etzkorn, L. 2006b. Indicating ontology data quality, stability, and completeness throughout ontology evolution. *Journal of Software Maintenance* **19**(1), 68–86.

Qi, G. & Hunter A. 2007. Measuring incoherence in description logic-based ontologies. In *Proceedings of 6th International Semantic Web Conference*, 381–394.

Schrimpsher, D. & Etzkorn, L. 2009a. Sub-graphing web service ontologies to support resource constraints of mobile devices. In *Proceedings of the 47nd Annual Association for Computing Machinery Southeast Conference (ACMSE2009)*, Clemson, SC.

Schrimpsher, D. & Etzkorn, L. 2009b. A Web service ontology sub-graph quality model to support mobile devices. In *Proceedings of the 3rd Annual International Symposium on Empirical Software Engineering and Measurement (ESEM2009)*, Orlando, FL.

Seidenberg, J. & Rector, A. 2006. Web ontology segmentation: analysis, classification and use. In *Proceedings of the 15th International Conference on World Wide Web (WWW2006)*, Edinburgh, UK.

Statistical and Process Management Software for Six Sigma and Quality Improvement–Minitab. 2010. *Statistical and Process Management Software for Six Sigma and Quality Improvement–Minitab*. State College, PA, Minitab Inc. http://www.minitab.com/en-US/default.aspx

Stuckenschmidt, H. & Klein, M. 2003. Integrity and change in modular Ontologies. In *Proceedings of the International Joint Conference On Artificial Intelligence (IJCAI2003)*, Acapulco, Mexico, **18**, 900–908.

Tartir, S., Arpinar, I., Moore, M. & Sheth, A. 2005. OntoQA: metric-based ontology quality analysis. In *Proceedings of IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, Houston, TX.

Volz, R., Oberle, D. & Studer, R. 2003. Implementing views for light-weight Web ontologies. In *Proceedings of the Seventh International Database Engineering and Applications Symposium (IDEAS2003)*, Hong Kong, 160–169.

Vrandecic, D. & Sure, Y. 2007. How to design better ontology metrics. In *Proceedings of European Semantic Web Conference (ESWC2007)*, Innsbruck, Austria.

Yao, H., Orme, A. & Etzkorn, L. 2005. Cohesion metrics for ontology design and applications. *Journal of Computer Science* **1**, 107–113.