On Boosting Semantic Web Data Access

Li Ding

January 13, 2005

ABSTRACT

The Semantic Web can be viewed as a collection of RDF graphs serialized by RDF documents that distributed in the Web. Its utility depends on three issues: availability (existence of data), accessibility (users can retrieve the data they want), and quality (users can judge the quality of the retrieved data). While more data are available in the Semantic Web, the latter two issues are often ignored or circumscribed due to lacking of tools and mechanisms. This dissertation proposes an ontology-based approach to these two issues so as to boost the utility of the Semantic Web.

For accessibility, we identified three critical challenges: i) there are few links to (and almost no description about) RDF documents; ii) it is hard to query the Semantic Web since users are not familiar with semantic web vocabulary (i.e. the URIrefs) with over 150,000 unique entries; and iii) it is unrealistic to query the entire Semantic Web without effective data access service. In order to address these challenges, we proposed the Web of Belief (WOB) ontology to model the Semantic Web and its context (i.e. the web and the agent world), and developed *Swoogle* system that digests and searches semantic web data using WOB ontology. In particular, Swoogle helps publishers by ranking properties of a given class, and supports information consumers by estimating query complexity and searching URLs of relevant RDF documents.

For quality, we first clarified the dimensions of quality (e.g. consistency, completeness, precision, importance and trustworthiness) for different concepts in WOB (e.g. RDF graph, web page, and agent). We then proposed the quality extension to WOB ontology for representing users' quality judgments (esp. trust judgments) explicitly. Finally, we proposes a series of semantic web navigation models and corresponding ranking algorithms for ontological terms and RDF documents, and a series of algorithms for evaluating trustworthiness of a given RDF graph according to the availability of background knowledge.

The contributions of this dissertation are the following: i) WOB ontology, which is one of the first attempts that make the Semantic Web self-descriptive in OWL semantics; ii) Swoogle, which is one of the first web-scale data access services that digest and search the Semantic Web; iii) semantic web navigation models and ranking algorithms; and iv) RDF graph trustworthiness evaluation mechanisms.

The WOB ontology and Swoogle like systems, we believe, will bring emergent properties to the Semantic Web: the utility of the web-scale Semantic Web will be reinforced when users are less hassled in finding useful data and are more aware of data quality.

CONTENTS

1. INTRODUCTION

> *"The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries"* (W3C Semantic Web Activity, http://www.w3.org/2001/sw/).

The Semantic Web can be viewed as an expressive, collaborative, and open information system in the Web for both intelligent agents and human users; hence it comes with the following features:

- **simple but expressive data model**. RDF graph data model uses URI-based vocabulary and RDF triples to describe the world.
- **collaborative publishing**. Agents publish data independently using common meta-ontology (i.e. RDFS and OWL) and same RDF graph model.
- **open system in web context**. Semantic web data is published throughout the Web. Both URI-based vocabulary and RDF triples are distributed extensible.

This dissertation interprets the Semantic Web as a web based information system (see figure 1.1): i) it is built on a web of (static or dynamic) RDF documents, each of which serializes an RDF graph and can be accessed in the Web by URL; ii) agents use Semantic Web data access service to obtain data encoded in RDF graph. We will use this interpretation throughout this dissertation.

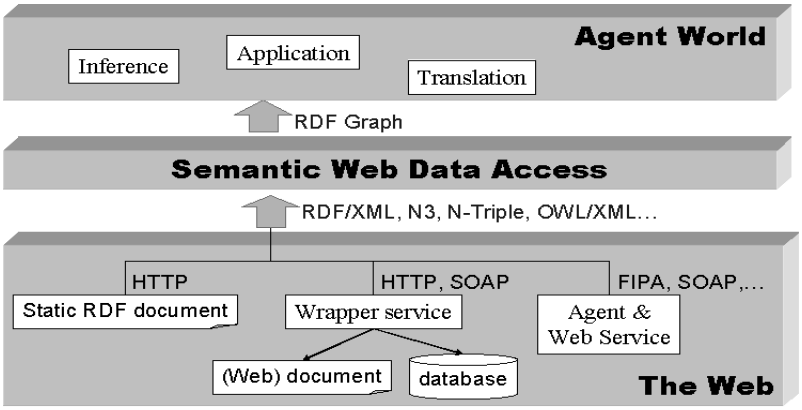

Fig. 1.1: The Semantic Web as a web based information system

1.1 Motivation

Our preliminary investigation (Swoogle) [25] has partially confirmed the existence and growth of the Semantic Web in the Web. Swoogle has discovered over 300,000 *RDF documents*[1] including over 4,000 *ontologies*[2]. Although the amount of RDF documents is far less than that of web pages[3], we did observe over 46,000,000 RDF triples. In addition, figure 1.2 shows the steady (or increasing) growth rate of RDF documents and ontologies[4]. Apart from the growth of semantic web data, many useful semantic web tools (e.g. JENA[5], Sesame [6]) have been developed to facilitate semantic web practices (e.g. building knowledge base). More evidences will be elaborated in appendix A.

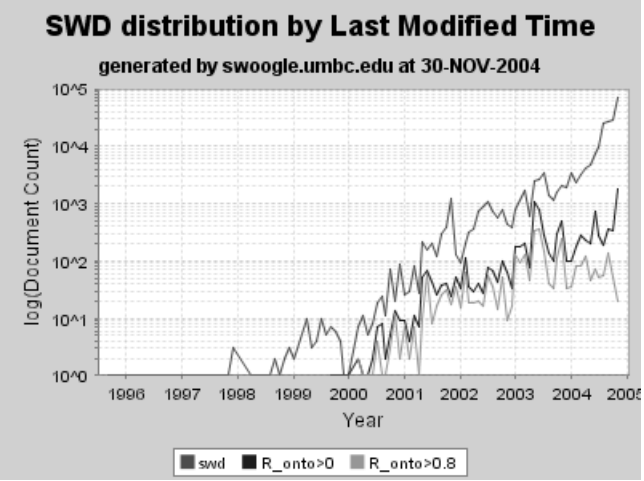

Fig. 1.2: The growth rate of the Semantic Web

The significant amount and steady growth rate of semantic web data lead to a question – *Is the Semantic Web a useful information system for web users?*

The utility of the Semantic Web can be best demonstrated in semantic web data access activities. A good lesson learned from *Usenet*, which " has grown dramatically but has become almost useless because of decreasing average quality"[40], shows that the utility of an information system depends on not only **(data) availability** (i.e. how much semantic web data is available in the Web) but also **(data) quality** (i.e. how good can semantic web data satisfy users' requirements). In addition, the success of web search engines (e.g. Yahoo, Google) showed us the third important factor – **(data) accessibility** (i.e. how easy and effective can users obtain the data they want).

[1] *RDF document* is equivalent to 'Semantic Web Document'(SWD), which is used by Swoogle.

[2] An *ontology* refers to an RDF document that has defined instances of *rdfs:Class* or *rdf:Property*.

[3] Google had discovered over 8,058,044,651 web pages by Dec 12, 2004

[4] R_{onto} refers to ontology ratio, which tells the proportion of defined classes and properties in all defined URIrefs. Swoogle use $R_{onto} > 0.8$ to select 'pure' ontologies.

[5] http://jena.sourceforge.net/

[6] http://www.openrdf.org/

The deployment of the Semantic Web has been hindered by the dilemma in ontology engineering world: *ontology developers want the others to adopt ontologies they created but seldom adopt ontologies created by the others*. We attribute this dilemma to many reasons: conflict interests in standardization, parallel research (e.g. RDF and Conceptual Graphs[98]), different domain expertise and focuses (e.g. DAML time ontology[45] and SOUPA[17] time ontology), lacking of trust and etc. The victims, unfortunately, are ontology consumers, who often get lost when facing vast amount of ontologies and get confused in choosing appropriate ontologies. This dilemma remarks the importance of enhancing *accessibility* and *quality* in semantic web data access.

Equation 1.1 summarizes three important factors for evaluating the utility of semantic web data access: data *availability* which depends on publishers' efforts and interests; data *accessibility* which depends on both data organization and data access services; and data *quality* which depends on information consumers' preferences and judgments.

$$Utility_{SWDA} = Availability + Accessibility + Quality. \tag{1.1}$$

Current semantic web practices approach the *availability* factor by producing data *manually* (using editors) or *automatically* (using information extraction tools). However, the latter two factors are not well addressed. *Accessibility* is mainly discussed in memory or database storage context; however, web-scale data storage and access mechanisms remain ignored. Perfect *quality* is often assumed in the absence of parser failure or semantic inconsistency. Although recent discussions on semantic web trust layer bring about more trustworthiness considerations such as authenticity (via digital signature) and trust-assertion (via belief and trust statement), systematic analysis and practice are still needed.

1.2 Challenges

The Semantic Web brings freedoms as well as challenges. This dissertation concentrates on challenges related to *accessibility* and *quality*.

Semantic Web Vocabulary. By *"using an extensible URI-based vocabulary"* [66], *"Everything can be identified by URIs"* [67]. This feature brings the freedom of referencing 'things' in real world and the challenge of managing the substantial amount of URI vocabulary. One 'thing' might be referenced by many URIrefs due to publishers' limited knowledge, e.g. the term "person" has been defined under more than 160 different namespace. In fact, the URI-based vocabulary simplifies the process of identifying a thing but complicates the process of constructing a query, e.g. how could a user compose a query that searches for instances of *person*, which corresponds to so many URIrefs (plus she might not even know any of these URIrefs). Moreover, many existing RDF statements are annotative (i.e. the object of an RDF statement is literal) and are only human-understandable.

Web-scale Semantic Web Data Access. By providing *"A Simple Data Model"*[66], *"Resources and links can have types"* [67]. This feature brings the freedom of data independence and the challenge of RDF graph web storage management. Existing works limit scope in managing RDF graphs in memory or database (e.g. JENA, Sesame,

Kowari[7]), and web-scale RDF graph storage management is often ignored. It is notable that RDF documents are poorly linked in compare with web pages since the Semantic Web lacks of concepts like 'hyperlink' in the Web.

Quality of RDF Graph. Since *"Anyone Can Make Statements About Any Resource"* [66], *"Partial information is tolerated" and "there is no need for absolute truth"* [67]. This feature brings the freedom of *distributed extensibility* (i.e. publishers are free of publishing RDF statements in the Web independently) and the challenge of dealing with open context and handling inconsistent/incomplete data. Current semantic web practices address inconsistency by validation, and often assume perfect information due to the absence of context information.

Both *accessibility* and *quality* are especially critical to *inference*, which has been launched as the second phase of the Semantic Web by W3C in February 2004. Logical inference mechanisms require relevant data in knowledge base and assume high quality data. Therefore, a reasoner, when processing an RDF document which did not mention the URLs of relevant ontologies explicitly, will first search relevant ontologies for URIref definition and then decide whether or not accept the ontologies according to their quality.

1.3 Thesis Statement

This dissertation will show that our ontology based approach is effective in building semantic web metadata and boosting web-scale semantic web data access with respect to accessibility and quality.

[7] http://www.kowari.org/

2. RESEARCH DESCRIPTION

2.1 General Description

This dissertation focuses on boosting web-scale data access activity in the Semantic Web. To this end, we first model the Semantic Web and its context with an ontology so as to make the semantic web self-descriptive. Then, we address accessibility issues by designing a web-scale semantic web data access service (*Swoogle*) for discovering, digesting and searching RDF documents in the Web. Finally, we address quality issues by developing semantic web quality evaluation mechanisms using navigation semantics and trust semantics.

2.2 Modeling the Semantic Web and Its Context

2.2.1 A Multi-World Conceptualization

The Semantic Web usually refers to the RDF graph world. Its context includes the Web and the agent world: the Web materializes the Semantic Web by serializing RDF graphs in RDF documents; and the agent world consists of human users and software agents who produce and consume RDF graphs. We should also be aware of the associations within and between these worlds: RDF documents are interconnected, e.g. an RDF document may reference several ontology documents for term definition; agents are interconnected by social relations like *foaf:knows*; and an RDF graph's provenance includes the RDF documents that serialized it, the agents who created it and etc. Therefore, the Semantic Web and its context can be interpreted as three interactive worlds: the Web (i.e. the web of online RDF documents), the RDF graph world (i.e. the web of RDF resources and literals), and the agent world (i.e. the social network of users). We will build an Web of Belief (WOB) ontology to materialize this multi-world conceptualization with the following objectives:

- It should capture important concepts and associations in the three worlds. A partial list includes: *agent*, *RDF document*, *RDF graph*, *association* and *provenance*.
- It should specify how concepts are uniquely identified and populated in the Semantic Web.

- It should be designed and populated within RDF. The ontology should be written in semantic web languages (i.e. RDF and OWL) and its instances should be stored in RDF graph.
- It should be defined in OWL DL for inference tractability purpose. Since many popular ontologies are based on RDFS, we may need to compose an OWL DL version for them.
- It should be *simple* and *clear* so that users may understand its semantics and use it in real world easily.
- It should reuse terms from existing ontologies as long as they have appropriate semantics and invent new concepts with reasonable justification. Also, we should be aware of how a concept relates to other existing terms and how it can be populated in the Semantic Web.

Related Work

Agent is a widely used concept: psychology, social science, and management science concern more about *human agent* and *community*; computer science (esp. artificial intelligence) concerns more about *software agent* [57, 94] and its applications in various fields [100, 56, 41]. In the Semantic Web, we have found that *agent* has been defined as class or property under over 100 different namespaces [1]. The most popular one is from *Creative Commons* ontology[2]. The most popular definition of *Person* comes from *Friend Of A Friend* (FOAF) ontology[3]. Unfortunately, they are semantically disconnected in the RDF graph world.

RDF document is a unique and important notion in the Semantic Web. In the Semantic Web, we have found some relevant terms: *foaf:Document* is popular but does not differentiate RDF documents from other online documents; and *owl:Ontology* normally references the ontology itself and does not help discover new RDF documents. There are also some syntax-oriented concepts in RDF Test [4] and OWL test[5], e.g. *RDF-XML-Document*,*N3-Document*, and *NT-Document* [6]. However, these terms do not capture the exact sense of *RDF document*. Note that this concept also helps building *hyperlinks* in the Semantic Web.

RDF graph is a complicate concept and section 2.2.2 details our concern.

Association is defined as "a relation resulting from interaction or dependence" by *wordnet16:Association* [7]. It is interchangeable with *relation*. A well-known definition of relation, *dc:relation*, comes from Dublin Core. We have two concerns regarding to this concept: i) *association arity*: RDF graph model is essentially binary association model, but how to represent higher-arity association? and ii) *machines-usable*: many popular properties such as *dc:source* are in fact annotative and not suitable for machine

[1] Swoogle has retrieved 113 different URIrefs which has *agent* as local name.(2004-12-16)
[2] http://web.resource.org/cc/
[3] http://xmlns.com/foaf/0.1/
[4] http://www.w3.org/TR/2004/REC-rdf-testcases-20040210/
[5] http://www.w3.org/TR/2004/REC-owl-test-20040210/
[6] These concepts are under namespace http://www.w3.org/2000/10/rdf-tests/rdfcore/testSchema
[7] source http://xmlns.com/wordnet/1.6/Organization

process; however, machines need properties defined in OWL DL to ensure inference tractability.

2.2.2 RDF Graph Reference

RDF graph reference is needed in making statements about *RDF graph*. Instead of embedding RDF graph, it describes how the referenced RDF graph can be uniquely obtained. It is needed in many situations such as i) annotating the provenance of an RDF graph, ii) representing logical relations between RDF graphs, e.g. *entails* and *supports*; iii) supporting agent belief statement; and iv) representing the premise and the conclusion of rules as well as a proof. Existing approaches are listed as the following:

- **Naive approach.** Naive approach references an RDF graph by the URL of an RDF document which serializes it. It is commonly used in semantic web practices, especially in testing semantic web reasoner (i.e. RDF Test, OWL Test) and in searching the Semantic Web (e.g. scutter[11], Swoogle crawler[25]). The limitation of this approach is that users can only reference the entire RDF graph serialized by the given RDF document – no more, no less.
- **RDF reification.** RDF reification [29] lets users reference RDF statement; however, we should be aware of that " the connection between the document and its reification must be maintained by some means external to the RDF graph syntax"[29]. Its semantics is under continuous discussion in RDF interest group [75, 29]. One of its limitations is that users can reference only one RDF statement with non-trivial space overhead.
- **Named Graphs** Recently, Named Graphs [16] is proposed to group a set of RDF triples into an instance of *RDF graph* and assign an URIref as *rdf:ID*; however, it extends the semantics and syntax of RDF. Named Graphs helps putting multiple RDF graphs in one document; however, is this advantage necessary? The limitation of this approach is that it is beyond RDF semantics and requires special syntax language (i.e. Trix[8]). In addition, both *naive approach* and *Named Graphs* allow only publishers to assign URIrefs to RDF graphs.

All these approaches require additional support to their semantics; and none of them is both efficient and expressive to reference an arbitrary RDF graph. Hence, an *RDF graph reference* language is needed.

2.2.3 Provenance

Provenance has been studied in digital library (e.g. Dublin Core[9]), database systems, (e.g. data provenance [13] and view maintains [19]) and artificial intelligence (e.g. knowledge provenance [21, 33] and proof tracing [20]). It refers to "the place of origin"

[8] http://swdev.nokia.com/trix/TriX.html
[9] http://dublincore.org/

according to WordNet2[10]. In the Semantic Web, provenance has finer meanings, e.g., "the RDF document that contains a certain statement", "the creator of a web page", "the web page that defines a term" and "the premise of a statement". Besides annotative usage, provenance data can be used in trust inference (e.g. trust propagation[91, 26]) and grouping RDF statements for search[11].

In the Semantic Web, *provenance*, *source* and *origin* are used interchangeably. *provenance* is defined by *Dublin Core Term* as a property to track ownership history, and defined by *WordNet1.6* as a subclass of *wordnet16:Origin* [12]. Note that *dc:source* is better recognized and used[13]; but its range is normally URL or the name of an agent. Other related terms are *dc:creator*, *dc:publisher*, *foaf:owner* and etc. Logical provenance was considered when representing proof trace, e.g. PML [20] and TRELLIS [36]. These approaches place loose constraints over the domain and range of provenance relation; moreover, they are too general to capture finer semantics of provenance in the Semantic Web.

2.3 Digesting and Searching the Semantic Web

A typical web-scale semantic web data access process is depicted in figure 2.1. The process involves interactions among three entities: a user, a data access service, and the Web. A **user** accesses semantic web data with three steps: i) composing query with semantic web vocabulary, ii) composing a local RDF graph by searching relevant RDF documents in the Web, iii) querying local RDF graph for information. To serve the user, a web-scale semantic web **data access service** should *discover* and *digest* RDF documents in the Web, and then provide search/navigation services for both RDF documents and ontological terms.

This dissertation focuses on building a web-scale semantic web data access service with the following functionalities:

- **Discovery.** It discovers/revisits semantic web data in the Web automatically. The automated RDF document discovery agent should be adaptive so as to be effective over time.
- **Digest.** It digests the metadata (annotations and relations) of the Semantic Web and its context.
- **Search and Navigation.** It concerns about query representation, query processing and navigation models. Search is different from navigation: users *search* with keywords and expect a list of matches; users start from a given RDF resource/document and *navigate* to another RDF resource/document via certain semantic links.

[10] source: wordnet2, http://www.cogsci.princeton.edu/cgi-bin/webwn (2004-12-18)

[11] provenance and topic are two useful ways to group data and thus support similarity related heuristics.

[12] source: http://xmlns.com/wordnet/1.6/Origin. However, the class *Origin* is defined confusingly due to its embedded class hierarchy: *Orgin* rdfs:isSubClassOf *Point*, *Point* rdfs:isSubClassOf *Location*, and *Location* rdfs:isSubClassOf *Entity*. It is hard to agree that *Origin* is a geographical point

[13] Swoogle reported that *dc:source* is defined by 55 ontologies,populated by 4282 documents and 17023 class instances.(2004-12-18)

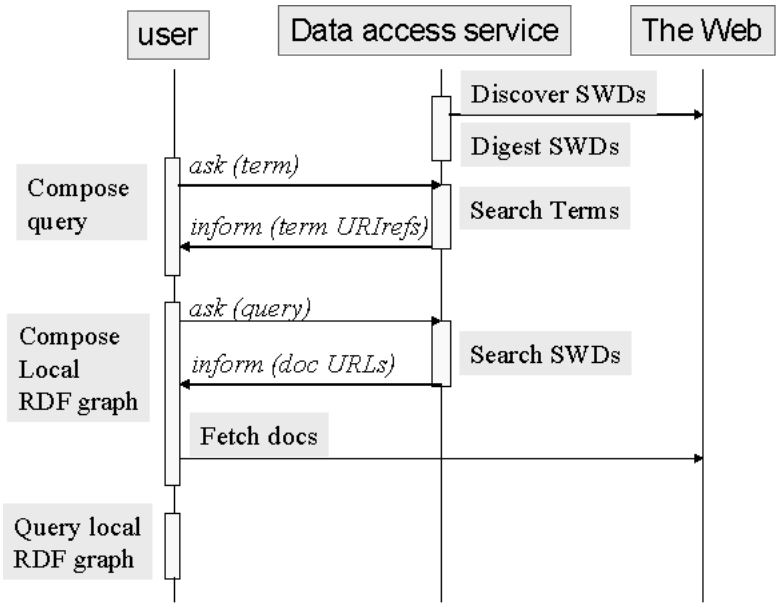

Fig. 2.1: A typical semantic web data access process

- **Rank.** It orders RDF documents and resources by their importance according to corresponding navigation models.

We will detail the first three objectives in the remainder of this section, and leave the last one in section 2.4.2 which discusses semantic web quality.

Related Work

From data access perspective, the Semantic Web lies between Information Retrieval (IR) systems and Database systems: its structured data model and data independence feature make it database systems alike, but its open framework and web-based data storage make it IR systems alike. Table 2.1 compares various aspects of data access among database, web IR and the Semantic Web.

There are also many similar data access systems in literature as the following:

- **Ontology based annotation systems**, such as SHOE [71], Ontobroker [23], WebKB [73], QuizRDF [22] and CREAM [42], focus on annotating and indexing online documents. Such systems organize metadata with their predefined ontologies and reason about data using the semantics provided by those ontologies; however, these systems do not differentiate RDF documents from text documents. It is notable that CREAM [42] had indexed 'proper reference' and 'relational metadata'.

- **Ontology repositories**, such as the DAML Ontology Library [46], SemWebCentral [48] and Schema Web [49], do not automatically discover RDF documents

Tab. 2.1: A comparison of data access services

	database	web IR	Semantic Web
storage	local file	webpage	RDF document
vocabulary	typed literal	strings	URIrefs, literal
data model	relation, table	free text	RDF Graph
implicit data	no	no	inferred triples
query optimization	table index	metadata	n/a
query language	SQL	keywords	RDQL, SPAQL
query scale	close	close/open	close/open
query result	any	document URL	any, document URL

but rather require people to submit URLs. They only collect ontologies which constitute a small portion of the Semantic Web. In addition, they simply store the entire RDF documents.

- **Semantic web ontology browsers**, such as W3C's Ontaria [47], provide a searchable and browsable directory of RDF documents (including Ontologies). Ontaria stores the entire RDF graph of each harvested RDF document (normally ontology) and does not produce metadata.
- **Semantic web instance databases**, such as *Semantic Web Search* [50], index instances of well-known classes (e.g. *foaf:Person*, *rss:Item*). They are backed by a high capacity RDF database and use fairly fixed data structure.

All the above approaches have limited support to web-scale semantic web data access: their metadata are not well designed for RDF documents and they neglect the IR aspect of the Semantic Web.

2.3.1 Discovering RDF Documents in the Web

This issue arises since there is no convenient ways to find RDF documents in the Web. Since there are few hyperlinks between RDF documents, RDF documents can only be discovered by traversing the Web and validating each encountered web document. Since large amount of RDF documents are sparsely distributed in the Web, it is impossible to scan and validate the entire web. Therefore, we need an effective and efficient method to discover only URLs of possible RDF documents and filter out irrelevant web documents before validating them.

In literature, there are several crawler-based approaches: i) *Meta-search* has been used by *Meta-crawlers* [92, 93] to take advantage of web search engines to find a needle in a haystack; ii) *Focused crawler* runs heuristics search that focuses on certain topics from one or several starting web pages, e.g. *scutter* [11] runs focused search to discovers URLs of possible RDF documents in RDF graph. The key issue with these approaches is to find the indicators that differentiate RDF documents from other documents, and we aim at a crawler with an adaptive indicator learner.

Besides first-time discovery, URL revisiting policy should be considered since it keeps the semantic web metadata up-to-date. In particular, we should be aware of

the following cases: i) the would-be RDF document which is intended to be an RDF document but not a valid RDF document yet, e.g. RDF document with minor syntactic errors; ii) the out-of-date RDF document which has been modified since last visit, e.g. RSS feeds.

2.3.2 Digesting the Semantic Web

We extend and populate the WOB ontology to summarize the Semantic Web with small amount of metadata.

Digesting RDF document focuses on building meta-description of a given RDF document D including: i) the annotative document properties of D; ii) the abstract of RDF graph serialized by D; and iii) the document-document relation and document-resource relations from/to D. Most existing works concentrate on the first one: i) Dublin Core focuses on building index for a digital library, ii) W3C's Annotea [14] Project focuses on bookmarks in the Web, and iii) RDFS and OWL define some annotation properties without specifying domain and range. Currently, Swoogle[25]'s metadata covers the first and the third aspects, and we are working on abstracting RDF graph.

Digesting term focuses on building a comprehensive definition of an *ontological term* which is the URIref of an instance of either *rdfs:Class* or *rdf:Property*. The semantic web vocabulary consists of URIrefs and literals. While literals are either typed literals (*rdf:XMLLiteral*) or strings, URIrefs constitute the unique semantic web vocabulary which is beyond natural language. We focus on **ontological terms** since they are well used as the basis of information sharing. Provenance of term definition should be digested since it helps users to locate term definition in the Web. Class-property bond[15] should be digested since it helps publishers to compose class instances. Term usage should also be digested for selecting the most popular terms[16].

2.3.3 Searching and Navigating Terms and Documents

Search service does not limit to keyword search, which is commonly used in IR for searching text documents. It should support both keyword search and structured RDF query and return a list of relevant RDF documents (or ontological terms). **Navigation** can be interpreted as canned search, which searches relevant terms or RDF documents for a given RDF document (or ontological term). E.g. users may want to navigate from a given term to terms having the same namespace.

Term search service helps users to find relevant **ontological terms**[17] for composing queries. Although providing term search service, W3C's Ontaria [47] is limited in its ontological term vocabulary and supported search/navigation mechanisms. We need to index more ontologies and provide richer search/navigation mechanisms:

- *Term lookup* searches terms by keyword

[14] http://www.w3.org/2001/Annotea/

[15] We currently only consider direct bond, which does not involve class inheritance.

[16] We currently only consider direct class-instance relation without any class inheritance involvement. So does property

[17] We will use 'term' to refer 'ontological term' in the following context

- *Term Index* lists all terms in alphabetical order of term prefix.
- *Similar Term* runs canned query to search terms sharing namespace, local name or provenance of a given term.
- *Term Definition* links a term to RDF statements that defined it.
- *Class-Property Bond* links a class to all its instance properties according to either ontology definition or instance statistics.

Ontology search service helps users to find ontologies in the Web to deal with *external reference* situation, i.e. an RDF document may reference an ontological term without specifying where to find its definition. This situation is caused by the dual semantics of the namespace of URIref, i.e., it both helps identify an RDF resource and shows the URL of ontology that defines the RDF resource. However, we can not assume the latter semantics since it is often violated in semantic web practice.

RDF document search service. Many semantic web users have mentioned the desire of searching the entire Semantic Web, regardless of ontology or instance data file. To address the demand, we should be aware of users' different interests in ontology documents and instance documents.

2.4 Evaluating Semantic Web Quality

The definition of quality as "fitness for use" ties *quality* with *utility*. Users' adoption of the Semantic Web as information source highly depends on whether users can judge data quality of the Semantic Web. *Data quality* (or information quality), which originates from business management, has attracted many researchers from Information Science and Computer Science [102, 99].

This dissertation focuses on two issues of semantic web quality:

- **To identify dimensions of semantic web quality**. According to [63], there are two categories of data quality in information systems: data product quality and data service quality. We are particularly interested in data product quality (which focuses on the content of data), which is orthogonal to data service quality (which focuses on data access service), and data authenticity (which focuses on security mechanisms in data access).
- **To evaluate semantic web quality**. We are particularly interested in i) ranking RDF resources/documents for comparing their importance and ii) evaluating arbitrary trustworthiness of an RDF graph or an agent for knowledge expansion. In addition, the amount of background knowledge and assumptions to be used in evaluation leads to different evaluation models.

We detail these issues and their related work in the remainder of this section.

2.4.1 Identifying Dimensions of Semantic Web Quality

The dimensions of *data quality* (e.g. accuracy, reliability, timeliness, accessibility) has been revealed by different approaches (e.g. intuitive understanding, experience based learning) in literature [102, 101]. Wang et al.[102] extensively surveyed dimensions of data quality and reported the following observations: *i)* database *researchers focused on data accuracy in terms of semantic and physical integrity; ii)* information system *researchers were aware of multiple dimensions of quality in information system but did not achieve consensus (e.g. Ballou et al. defined dimensions of quality as accuracy, timeliness, completeness and consistency; Wang et al. identified categories of quality as intrinsic, contextual, representation, and accessibility; other dimensions are data validity, availability, tractability and credibility); iii)* psychology *researchers focused on user satisfaction requirements over information system (e.g. Kriebel identifies accuracy, timeliness, precision, reliability, completeness and relevancy); iv)* auditing *methods brought about statistical measures such as frequency, distribution to quantify accuracy; v)* ontological-based *method [101] modeled the world and information system as states and laws, and thus came up with more accurate dimensions such as incompleteness, ambiguous, and meaningless; and vi) some researchers also adopted information-theory-based or marketing-research-based approaches.*[18] Wand and Wang [101] further ranked data quality dimensions by citations: the top five dimensions (i.e. accuracy, reliability, timeliness, relevance, completeness) were much better cited than the rest. Based on the above work, Wang and Strong [103] listed 16 dimensions of data quality.

Most of the above data quality dimensions are rather intuitive and ambiguous. For example, *accuracy* is defined by "the correctness of the output information"; however, *correctness* is still unclear to users. The ontological approach [101] seems to be promising; however, its state world abstraction might be unrealistic, and its model only covers some data quality dimensions. Besides, we should be aware of different semantics of semantic web data, e.g. RDF resource and RDF document are conceptual entities but RDF graph is rather knowledge statement. Therefore, we need to identify different quality dimensions for different type of semantic web data.

2.4.2 Ranking ontological terms and documents

Ranking ontological term. When publishing information, users may want to use the most popular ontological terms. For example, I would rather publish my personal profile using FOAF ontology than other ontologies (even they are composed by me). The importance of a class C depends on many factors, such as i)how many C's direct instances are on the Semantic Web[19], ii) how many RDF documents has populated C', and iii) what properties are normally used to modify C's instances. The importance of a property depends on i) its usage as *rdf:predicate* in RDF triples, ii) its usage in RDF documents and etc.

Ranking RDF document. RDF document has two unique features: i) it differs from text document since its vocabulary contains both URIrefs and natural language

[18] These are rephrased content from the original article.

[19] Inheritance relation may introduce indirect instances and thus complicate the problem.

text and ii) it differs from HTML document since there are few hyperlinks between two RDF documents and RDF documents are often interlinked by sharing URIrefs. Therefore, *ranking RDF documents* remained untouched until our previous work [25], which built hyperlinks between RDF documents and ran a simple PageRank variation to compute ranks. Although recognized the importance of link semantics, that work only consider the web of RDF documents and ignored document-resource relations. An important application is to rank ontologies by their importance and to recommend the best ontology to inference engine.

Related Work

- **complex network analysis**. It focuses the structure properties of a direct graph and attempts to reveal the overall quality of a graph. Hence quality is interpreted as clustering coefficient, degree distribution, average path length and etc. Newman provided a comprehensive survey [79]. In addition, we may construct a weighted directed graph based on the frequency of URIrefs associated with edges.

- **text document ranking**. It focuses on ranking text documents with their importance according to their content. Vector-space model is commonly used in conventional IR to determine document similarity. TFIDF [90] plays an important role in ranking documents.

- **web page ranking**. It ranks web pages using their unique hyperlink structure. Graph navigation interpretation is commonly used in web IR, and there are some well-known ranking methods [70] such as PageRank[80, 44], Hits [64].

- **semantic ranking**. It ranks resources in an RDF graph according to semantics of links. Zhuge and Zheng [55] extends PangRank algorithm with a non-uniform surfing model, where RDF semantics are used to assign weights to links.

- **social network analysis**. It focuses on structure properties of a social network and attempts to rank nodes in a graph. Hence quality is interpreted as degree centrality, closeness centrality and betweenness centrality [43].

The above approaches reveal the important factors in ranking: content, graph structure and semantics of links; however, none of them fully utilizes the unique semantics with RDF.

2.4.3 Evaluating RDF Graph Trustworthiness

While rank focus on nodes in RDF graph, trustworthiness studies edges in RDF graph. RDF graph trustworthiness lies in users' beliefs about the quality of RDF graph as follows:

- **content-based quality**. Users need to known semantic consistency with respect to background semantics. Users need to compare two RDF graphs to determine

their content similarity (e.g. implication)[20] and difference [5].

- **complexity**. Users need to estimate inference complexity of a set of RDF document beforehand.
- **context-based quality**. Users may judge RDF graph credibility based on background information such as provenance and trust.

Related Work

Trustworthiness is an important branch of context-based quality. The *semantic web trust layer* first appeared in W3C's layer cake of the Semantic Web (figure 2.2).

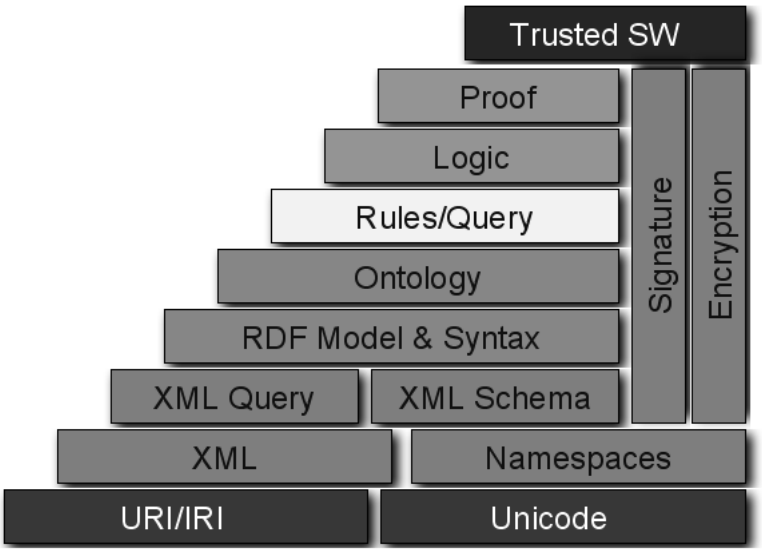

Fig. 2.2: The semantic web layer cake.
(graph source: http://www.w3.org/2004/Talks/0209-Helsinki-IH/105.html)

The original idea concerns the trustworthiness of information to deal with the reality of "no absolute truth" and evaluates trustworthiness using peer annotation mechanisms secured by digital signature[54]. Recently, researchers [37, 89, 88] have remarked the important role of "trust network" as an social alternative to conventional security approaches in propagating trust. Carroll and Bizer summarized the trust layer with three types of trust mechanisms (namely reputation-, context- and content-based) in semantic web publishing context [15]. SWAD-Europe also delivered a series of proposals to encode trust [3]and recommendation into the Semantic Web. Although these proposals did not reach the consensus of the exact requirements of trust layer, they

[20] Implication plays important role in determining relative precision. For example, given *(foaf:Person, rdfs:subClassOf, foaf:Agent)* the RDF graph *(foo:Li , rdf:type, foaf:Person)* is more precise than *(foo:Li, rdf:type, foaf:Agent) since it imply the latter*.

share similar ideas: i) trust and provenance are the keys and ii) trust representation and computation for propagating beliefs in open network are the main tasks. Unfortunately, most of these works remain in theoretical discussion and use vague and general trustworthiness definition. In fact, enforcing trust mechanisms in the Semantic Web requires non-trivial efforts.

2.4.4 Trust and Provenance based Navigation

Trust and provenance information offers semantic web users a new navigation mechanism: a user may incrementally expand her knowledge base by navigating the web-scale Semantic Web. There are two well-known heuristics: *similarity heuristic*, which favors information sources having similar domain interest, and *trust and provenance based heuristic*, which favors trusted information sources. The primary difference between the two is that the former requires a centralized index service that compares all information sources while the latter runs in a peer to peer fashion.

3. RESEARCH PLAN

3.1 Research Methodology

This dissertation addresses semantic web data access problem using divide-and-conquer method, and divides it into three related sub-problems. For each sub-problem, we solve it in the following steps:

1. Identify and describe the problem with requirements
2. Review related work
3. Propose evaluation method
4. Propose and implement approach
5. Assess approach
6. Show the contribution

3.2 Research Objectives

Modeling the Semantic Web and Its Context

We designed Web Of Belief (WOB) ontology for modeling the Semantic Web and its context in terms of three interactive worlds.

1. Identify concepts and associations in the Semantic Web and its context.
2. Build a core ontology focusing on the most important concepts and associations, and show how they can be uniquely identified and used. We remark two important concepts as follows:
   - *RDF graph reference language*. We will differentiate our work with the *Named graph* approach, and show how it could be used.
   - *Provenance*. We will refine provenance relations in the Semantic Web and bring more formal semantics for inference purpose.
3. (TODO) Show this ontology can be instantiated through mapping and rules [1].

[1] Simple text processors may be needed to standardize an input RDF document. For example, we need to differentiate URL from name when processing the text within *dc:creator* tags.

The WOB ontology will be evaluated by the following methods:

1. Analytical comparison with other existing ontologies.
2. Statistical report on the usage of ontology, e.g. (direct and inferred) instance of concepts. We should note whether the instances are good for inference or annotation.

The deliverable artifacts are the follows:

- WOB core ontology with a descriptive document.
- OWL DL version of popular Ontologies such as FOAF and DC.
- RDF graph reference language specification.
- The translated instances of WOB core ontology.

Digesting and Searching the Semantic Web

Aiming at *accessibility* issue, we developed Swoogle to provide a web-scale data access service for semantic web users.

1. Discover URLs of RDF documents in the Web.
2. Digest Semantic Web metadata. It is highlighted by WOB extension for RDF document metadata and RDF graph abstract. RDF graph abstract is one of the highlights of this dissertation.
3. Provide search and navigation services as follows:
   (a) Search ontological terms by keywords.
   (b) Search ontologies by keywords. We are working on other search mechanism, e.g. 'search ontology for a given RDF graph', and 'search RDF documents for a SPARQL[2] query'.
   (c) Navigate terms and RDF documents according RDF semantics. Navigation models are highlights of this dissertation.
   (d) Estimate query complexity for a given SPARQL query.

Swoogle will be evaluated by the following methods:

1. Statistical report on collected metadata, e.g. how many RDF documents has been discovered, and how many terms are used by each RDF document. Swoogle statistics[3] will fulfill this job.
2. Real world usage of Swoogle. That includes semantic web researcher who use Swoogle under certain contract and public users. Swoogle website visit statistics could be a good indicator.

[2] http://www.w3.org/TR/rdf-sparql-query/
[3] http://swoogle.umbc.edu/modules.php?name=Swoogle_Statistics

3. Analytical comparison between Swoogle and other existing alternatives, which is mentioned in section 2.3.

The deliverable artifacts are the follows:

1. Adaptive RDF document discovery agent.
2. Semantic web metadata published in RDF.
3. RDF term/document search/navigation service via both web and web service interface.
4. Swoogle statistics of semantic web data and its users collected by Swoogle.

Evaluating Semantic Web Data Quality

Aiming at *quality* issue, we will clarify dimensions of semantic web data quality, and propose some evaluation metrics and mechanisms.

1. Clarify dimensions of semantic web data quality.
2. Evaluate data product quality
   (a) Ranking RDF resource and RDF document. It is highlighted by several navigation models using different amount of RDF semantics.
   (b) Evaluating RDF graph trustworthiness. It is highlighted by explicit representation and computation of trustworthiness.
3. Trust and provenance based navigation.

Swoogle will be evaluated by the following methods:

1. Analytical justification of quality dimensions.
2. Analytical comparisons between different ranking methods with respect to their underline navigation model, and convergence analysis.
3. The effectiveness of trust and provenance based navigation model.
4. Trustworthiness can be evaluated using analytical method, simulation[26], and user satisfaction in applications (e.g. SEMDIS).

The deliverable artifacts are the follows:

1. A list of semantic web data quality dimensions.
2. A series of ranking methods for RDF resource and RDF document.
3. A trust and provenance based semantic web navigation model.
4. A series of trustworthiness analysis methods for RDF graph.

3.3 Research Schedule

Tab. 3.1: Research time table

Phase	**Objectives**	**Artifacts to produce**	**Status** (%)	**Time** ($months$)
1	WOB			[**2**]
		WOB-core ontology	60	0.5
		OWL DL popular ontology	50	0.5
		RDF graph reference ontology	30	1
		translated WOB-core instances	0	1
2	Swoogle			[**6**]
	(discovery)	adaptive discovery agent	50	1
	(digest)	semantic web metadata	50	2
	(search)	search/navigation service	30	2
	(evaluation)	Swoogle statistics	30	1
3	Semantic Web quality			[**6**]
	(quality dimensions)	WOB-quality extension	20	1.5
	(rank)	ranking algorithms	30	2
	(navigation)	a navigation model	50	1
	(trustworthiness)	trust inference algorithms	50	1.5
4	Finalize	dissertation composition		[**4**]
Total				[**18**]

4. PRELIMINARY WORK: MODELING THE SEMANTIC WEB AND ITS CONTEXT

4.1 The Web of Belief Conceptualism

We developed the *Web of Belief* (WOB) conceptualism, which is named after one of W. V. Quine's book, to describe the concepts and relations of three worlds (see figure 4.1), namely the Web, the RDF graph world, and the agent world. Inter-world relations include provenance, e.g. who published an RDF document. Inner-world relations include RDF document reference, RDF graph reference, and social relations (especially trust) among agents.

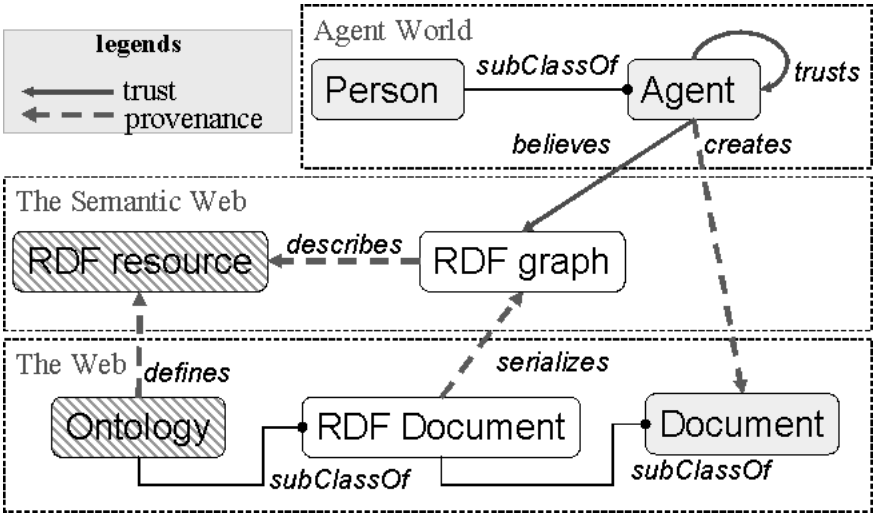

Fig. 4.1: The WOB conceptualism overview

4.2 Designing WOB core ontology

The WOB ontology is designed incrementally: the core ontology only covers a small set of stable and essential terms, and more terms will be added through extension ontologies.

4.2.1 WOB Core Concepts

As shown in figure 4.1, some of WOB core concepts reuse existing ontologies: the shaded nodes are from meta-ontologies and the gray nodes are from popular ontologies.

- *rdfs:Resource* is from meta-ontology RDFS.
- *owl:Ontology* is from meta-ontology OWL. It refers to an RDF document that defined instances of *rdfs:Class* or *rdf:Property*.
- *foaf:Agent* is from FOAF ontology. It could be a human agent, a software agent, or an organization. Although this term is less popular than *cc:Agent*[1], it comes with clear semantics [2] and has well-known subclasses including *foaf:Person* and *foaf:Organization*.
- *foaf:Person* is from FOAF ontology. It refers to a person.
- *foaf:Document* is from FOAF ontology. It refers to a web document.

However, we can not find appropriate URIrefs for two important concepts: *RDF graph reference* and *RDF document*.

We list some URIrefs relevant to *RDF document* as follows:

- `http://xmlns.com/foaf/0.1/Document`. It is populated by 15072 documents and 17265 class instances; however, it is so general that includes all web documents.
- `http://xmlns.com/foaf/0.1/PersonalProfileDocument`. It is populated by 8811 documents and 8819 class instances; however, it might be too specific since it is a "personal profile RDF document".
- `http://www.w3.org/2002/07/owl#Ontology`. It is populated by 3588 documents and 3804 class instances; however, it might be too specific since it only covers RDF documents defining ontological terms.
- `http://www.w3.org/2000/10/rdf-tests/rdfcore/testSchema#RDF-XML-Document`. It is populated by 1003 documents and 18010 class instances; however, it is only used in RDF test and OWL test and is syntax specific. We also find some similar URIrefs under the same namespace, such as *inputDocument*, *conclusionDocument*, *premiseDocument*,*NT-Document*, and *outputDocument*. There are other namespaces serving similar purpose, e.g. `http://www.w3.org/2003/03/rdfqr-tests/query.rdfs` and `http://www.daml.org/services/owl-s/1.0/Grounding.owl`.

[1] Swoogle reported that cc:Agent has been populated by about 3,000 documents but foaf:Agent has been populated by only 100 documents

[2] Our investigation showed that most instances of *cc:Agent* are in fact referring to persons. That means *cc:Agent* is functionally interchangeable with *foaf:Person*.

The above definitions are either too general or too specific, and the syntactic encoding should be better encoded as a property. Hence we propose **wob:RDFDocument**, which a subclass of *foaf:Document*. RDF documents refer to only those documents written in W3C recommended syntax (i.e. RDF/XML, N3, N-Triple), and embedded RDF are not considered for the time being. Its *rdf:ID* has two semantics: i) a URL through which users can retrieve the RDF document in the Web; ii) a URIref that uniquely identifies the RDF document.

Similarly, we list some URIrefs relevant to *RDF graph* as follows:

- `http://purl.org/puninj/2001/05/rgml-schema#Graph`. It is used for representing a geometric graph.
- `http://www.w3.org/2004/03/trix/rdfg-1/Graph`. It refers to an RDF graph. This URIref is used by named graph approach; however, it defines an instance RDF graph but not a reference to RDF graph.

These definitions are not suitable representing *RDF graph reference* and are poorly populated. Therefore we propose **wob:RDFGraphRef** to uniquely reference an arbitrary RDF graph for information consumers. An instance of *wob:RDFGraphRef* is not the referenced RDF graph, it only tells how to obtain the referenced RDF graph. We should also be aware that the referenced RDF graph may not necessarily exist in any RDF document. We will detail this concept in section 4.3.

4.2.2 WOB Core Associations

RDF data model uses instances of rdf:Property to capture the binary relation between two resources; but how could users make assertions about a relation and how to represent N-ary relation ($N > 2$). Therefore, we proposed **wob:Association** to represent N-ary relations. *wob:Association* has a necessary property *wob:connective* which names the association type.

Provenance is an important relation in WOB. We defined *wob:source* to represent generic provenance relation and listed its sub-properties in section 4.4.

4.2.3 Unique Instance Identity

Although URIrefs can uniquely identify things in the world, users may choose different URIrefs to refer the same thing. For example, how do we know that two instances of *foaf:Person* are referring to the same person. There are four types of commonly used identities:

- **institutional unique identity.** Some identities are guaranteed unique by their supporting systems, e.g. URL, email, mail address, phone number, user account in a particular system (e.g. MSN[3]), scientific taxonomy.
- **biological unique identity.** DNA and fingerprint are well-known unique biological identity.

[3] http://www.msn.com/

- **semi-unique identity.** Some identities do not guarantee collision-free but do demonstrate negligible collision possibility. Hash functions of unique-identity are used to protect privacy, e.g. mbox_sha1sum.

- **partial identity.** Some identities are used in our daily life but could refer to different things, e.g. name, and word. Some unique identities can be used as partial identities of person, such as phone number, and name. By combining multiple partial identities, we may reference a person with strong confidence. For example, credit card companies simply use name, address and phone number to locate a person.

Now we discuss how WOB core concepts can be uniquely identified.

- *wob:RDFdocument* and *foaf:Document* can be uniquely identified by their URLs.

- *wob:RDFGraphRef* references RDF graph and one RDF graph could be contained in more than one RDF documents. However, the instance of *wob:RDFGraphRef* does produce only one RDF graph.

- The identity of *foaf:Agent*, such as *organization* and *foaf:Person*, is a complicate issue. The identity of person highly depends on social convention: i) a person can be best identified by biological identity; ii) a person, if in the Web, can be best identified by her email, and then her homepage, msn account, and so on; iii) name normally does not uniquely identifies a person (even DBLP put multiple persons' publication under one directory); however, name may be the only identity information we can get in some context (e.g. the authors of publication). Currently, we use the *owl:ReverseFunctionalProperty* defined in FOAF ontology to uniquely identify instances of *foaf:Person* and *foaf:Agent*.

4.3 RDF Graph Reference

wob:RDFGraphRef shares the idea of *named graph* but focuses on embedding the concept RDF graph reference in RDF. An instance of *wob:RDFGraphRef* references an arbitrary RDF graph. The semantics of *wob:RDFGraphRef*, inspired by RDF query, considers different ways to construct an RDF graph:

1. *wob:sourceDocument*. An instance of *wob:RDFDocument* is the *wob:sourceDocument* of an instance of *wob:RDFGraphRef*. The RDF graph being referenced is the entire RDF graph serialized in that RDF document. We impose 0 or 1 cardinality constraint on this property.

2. *wob:usePattern*. The referenced RDF graph should contain a set of triples that satisfies *wob:TriplePattern*.

3. *wob:rejectPattern*. The referenced RDF graph should not contain any triple that satisfies *wob:TriplePattern*.

An instance of *wob:RDFGraphRef* is well-founded if it can produce a unique and valid RDF graph. Inconsistency may occur when i) *wob:usePattern* and *wob:rejectPattern* overlaps; and ii) the RDF graph referenced by *wob:source* does not satisfy *wob:usePattern*. Incompleteness may occur when i) nothing or only *wob:rejectPattern* is specified; and ii) *wob:usePattern* contains non-simple-triple patterns in the absence of *wob:sourceDocument*.

wob:TriplePattern shares ideas from RDF query languages but it focuses on extracting RDF graphs. Currently, we only support one of its subclass *wob:SimpleTriple* which references a triple like RDF reification does.

4.4 Provenance

In the semantic web, users have different concern about provenance. *dc:source* is widely used to annotate the provenance of a document or a resource. Let *aURL* be a URL, *aURIref* be the URIref of a class individual, *bNode* be the URIref of an blank node, and *literal* be an arbitrary literal. Below is a partial list of various usage of *dc:source* in triple format:

- *(aURIref, dc:source, aURL)* [4]
- *(bNode , dc:source, literal)* [5]
- *(aURL, dc:source, literal)* [6] [7] [8]
- *(aURL, dc:source, aURIref)* [9]
- *(aURL, dc:source, aURL)* [10]

dc:creator and *dc:publisher* are used similar to *dc:source*, and they are followed by usually the name(or nickname) of a person or sometimes the URIref of an instance of *foaf:Person*.

In addition to the scope of *dc:source* and *dc:creator*, WOB considers more subjects of provenance such as RDF graph and more sub-types of provenance.

As shown in figure 4.2, the provenance of an RDF graph can be refined to three sub-types:

- **where-provenance** shows the RDF documents that serialize a certain RDF graph. In WOB-core, it corresponds to *wob:sourceDocument*.

[4] source: http://www.peerfear.org/rss/permalink/2004/06/14/IHateMicrosoft/comments.rdf (2004-12-21,it is a would-be RDF document)

[5] source: http://derpi.tuwien.ac.at/ andrei/cerif/cos-prof.rdf. (2004-12-21,it is a would-be RDF document)

[6] source: http://www.ontoknowledge.org/oil/case-studies/CIA-in-OIL.rdf("to be added") (2004-12-21,it is an RDF document)

[7] source: http://www.planetrdf.com/index.rdf (a rss file, "Raw Blog of somebody") (2004-12-21,it is an RDF document)

[8] source:http://freepages.genealogy.rootsweb.com/ glennholliday/mundy/mundy_family.rdf ("original"). (2004-12-21,it is an RDF document)

[9] source: http://simile.mit.edu/repository/welkin/latest/data/Short.rdf. (2004-12-21,it is an RDF document)

[10] source: http://iswc2003.semanticweb.org/overview.rdf. (2004-12-21,it is an RDF document. W3C RDF validation reported error because of its XML charset setting, but it worked fine when the entire RDF part was pasted in its text input window.)

- **whom-provenance** shows the relation between publisher and RDF graph. In WOB-core, it corresponds to *wob:author* . This relation can be derived from the combination of *wob:sourceDocument* and *dc:creator*. The range is *foaf:Agent* since both *foaf:Person* and *wob:Website* can be the source agent.
- **why-provenance** associates RDF graphs with logical proof. It has strong inference background and we will discuss it in future work.

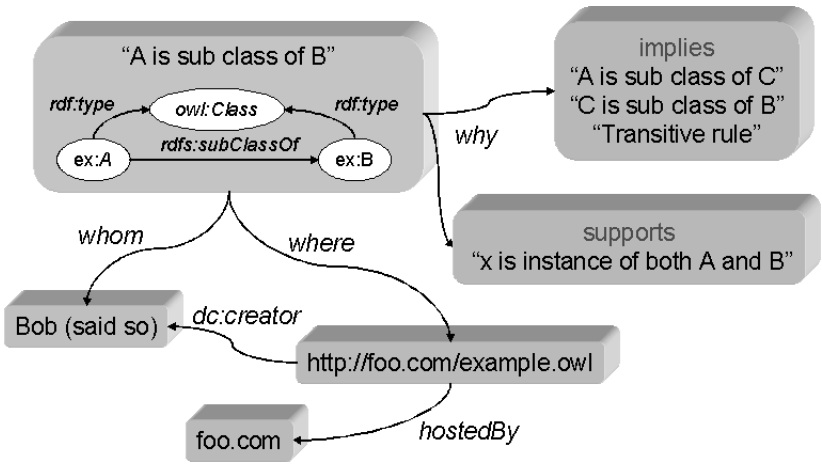

Fig. 4.2: Provenance of RDF graph

We also consider the provenance of ontological terms, i.e. where classes and properties are defined. *rdfs:isDefinedBy* is normally used to link to an human readable specification of a term. WOB uses *wob:isDefinedBy* to associates an ontological term with the *wob:RDFDocument* which contains its definition.

The current WOB provenance relations are shown in figure 4.3. It is notable the there are two kinds of provenance, the provenance relations in dark color are from existing ontologies and are normally used for annotation purpose; the provenance relations in light color are defined by WOB and can serve the inference purpose due to their clear semantics and *rdfs:range* constraints. These two types of provenance relations run in parallel.

The provenance relations defined by WOB are listed as below:

- **source**. It is the generic provenance relation.
- **sourceDocument**. It belongs to where-provenance type and points to a source document. It is the inference counterpart of *dc:source* and *rdfs:isDefinedBy*.
- **creater**. It belongs to whom-provenance type and points to an agent. It is the inference counterpart of *dc:creator*.
- **isDefinedBy**. It belongs to definition-provenance type and points to RDF document that defined/declared a term. In this sense it works similar to *rdfs:isDefinedBy* which points to an html explanation document.

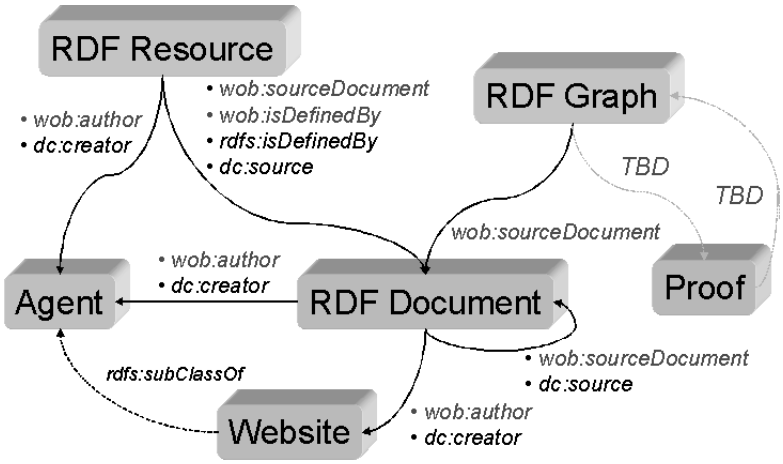

Fig. 4.3: Provenance in WOB

Note that why-provenance between *RDF graph* and *proof* is in future work list.

5. PRELIMIARY WORK: SWOOGLE – DIGESTING AND SEARCHING THE SEMANTIC WEB

5.1 Discover RDF Document in the Web

Discover RDF documents in the Web is critical in populating *wob:RDFDocument*. The sparse distribution of semantic web data in the Web forbids brute forth scan on the entire web. Therefore we consider several crawler based approaches as follows:

- A straightforward approach is to run meta-search on existing web search engines. Since Google is reported as indexing the largest portion of the Web, plus its support of *filetype* search, we built a **Google-crawler** to take these advantages. The key research issue here is to find *word indicators* of RDF documents.
- If knowing that a web directory contains many RDF documents, plus these documents can be reached from a given web page via hyperlinks, we may run a **focused-crawler** to do brute-forth search within that web directory. The focused-crawler is not topic-based since it only cares whether a web document is an RDF document.
- Some triples in RDF graph may also contribute URLs according to their semantics. We maintain an extensible list of such indicators in our **Semantic Web crawler** so that it can discover new URLs when processing RDF documents. Example indicators are i) namespace of a URIref, ii)(?urlRDF , rdf:type , owl:Ontology), iii)(_ , rdfs:seeAlso , ?urlRDF), and iv) (_ , owl:imports, ?urlRDF).

5.1.1 Find Word Indicators of RDF Document

Word indicators are used to find possible RDF documents and to exclude impossible RDF documents with conventional web search engine. We are interested in two categories of word indicators: i) *keywords* that differentiate RDF document from the others; and ii) *cat-words* that refine Google search [1]. We will discuss the second issue in future work.

Keywords

We first manually discover sub-categories of keywords and use them in search. Then, we will run informed inductive learning on discovered documents to find more key-

[1] Google return at most 1,000 results for any query

words. The learning result can be used in two folds: i) reinforce discovery process with more learned keywords, ii) estimate the possibility that a document is an RDF document.

Filetype. Many RDF documents follow RDF naming convention on file extension as shown below. Some RDF documents use *.xml* extension. However, we also found many documents with no extensions, some of which are dynamically generated upon query. According to Swoogle' statistics, only ten extensions (see table 5.1) could be used as keywords with accuracy higher than 50% [2]. $Recall$ shows the proportion of RDF documents using the given extension among all RDF documents. $Accuracy$ shows the proportion of RDF documents among all documents using the given extension.

	extension	**# RDF docs**	**Recall**	**Accuracy**
1	rdf	222,597	66.18%	72.93%
2	rss	20,356	6.05%	58.56%
3	xml	11,744	3.49%	31.09%
4	owl	5,042	1.50%	89.08%
5	daml	2,477	0.74%	77.87%
6	n3	1,433	0.43%	35.97%
7	nt	509	0.15%	74.20%
8	foaf	453	0.13%	96.80%
9	rdfs	416	0.12%	82.87%
10	xrdf	178	0.05%	88.12%
	'no-extesion'	70,875	21.07%	12.21%

Tab. 5.1: Extensions of RDF documents (Dec 23, 2004)

It is also notable that we can use file extension to exclude non-RDF documents. We have found over 200 extensions that indicate non-RDF documents in most cases. Table 5.2 lists some examples. $Accuracy$ shows the percentage of non-RDF documents to all discovered documents using the given extension.

URL pattern. Besides file extension, words in URL may help find possible RDF documents. For example, *foaf*, *owl*, and *rdf*.

TODO: We will build a classifier to find URL patterns keywords.

Content Pattern. Some words or phrases in a document can also help find possible RDF documents. For example, *rdf:RDF* and RDF namespace are good content pattern keywords. On the other hand, we can also exclude non-RDF documents with keywords, e.g. we found that "%22Template-type%22" is a keyword that indicate a class of non-RDF documents which have file extension *.rdf*.

TODO: Positive word/phrase list

TODO: Negative word/phrase list

TODO: Content based document classifier.

[2] although 'n3' does not satisfies this condition, we did find over 1,000 RDF documents with it

extension	# non-RDF docs	Accuracy
html,htm, *html	-	-(100%)
gif,jpg,jpeg,png	-	-(100%)
stm	11436	100%
php	1507	99.34%
asc	1403	100%
txt	1285	93.75%
asp	269	99.63%
pl	250	99.63%
cfm	223	91.67%
xsl	155	100%

Tab. 5.2: Extensions of non-RDF documents (Dec 23, 2004)

5.1.2 Revisiting Policy

After an RDF document is discovered, we need to revisit it to get the latest version. Some RDF documents, such as RSS documents, are frequently updated, and others, such as ontology, are more stable. Revisit strategies is studied intensively in web context [10, 30], and we are interested in revisit policy for the following types of documents in semantic web context.

The would-be RDF document. Some documents are intended to be an RDF document but rejected by semantic web parsers. By tracking these documents, we may learn to avoid common errors in composing RDF documents. In addition, we can revisit these documents since the author may have corrected them. The would-be RDF document can be identified by applying the classifiers described in section 5.1.1.

The out-of-date RDF document. Not all web documents are persistent in the Web – they may be updated or deleted without notification, so does the Semantic Web. One important research is to control revisiting frequency for RDF documents in different update rates. Another important point is to keep records for dead RDF documents.

The redirected RDF document. Redirected URLs are commonly used to maintain an evolving RDF document under a permanent URL. Such URLs should be revisited the same as URLs of valid RDF documents.

5.2 Digesting the Semantic Web

By digesting the Semantic Web, we are not limited in collecting annotative information; in stead, we would like to build more structured metadata for inference. The semantic web metadata includes not only the concepts such as RDF document, RDF graph and resource, but also the relations throughout the three worlds of WOB conceptualism. In this sense, we are building an RDF graph that abstracts the entire Semantic Web.

5.2.1 RDF Document Annotation

Our previous work [3] focuses on annotating the document properties of RDF documents in three levels.

Document level properties only cover annotative properties of a document.

1. *swoogle:suffix*. i.e. the file extension of the document.
2. *swoogle:discoveredDate*. i.e. when the document is discovered.
3. *swoogle:lastmodifiedDate*. i.e. when the document is last modified. It also imply the age of document.
4. *swoogle:md5hash*. i.e. the md5 hash code of the document. It can be used to compare if two RDF documents are identical before comparing document content [4].
5. *swoogle:reachable* . i.e. whether the document can be reached by web crawler.

RDF/OWL level properties cover the properties uniquely possessed by an RDF document. Besides Dublin Core annotative properties (e.g. *dc:creator*, *dc:publisher*, *dc:title*, *dc:date*, *dc:language* and *dc:subject*), we list the properties that can be used for both annotation and inference.

1. *wob:useRDFsyntax* shows the syntactic encoding of an RDF document. There are three existing encodings, namely "RDF/XML", "N-TRIPLE" and "N3". It is rather a MUST property of *wob:RDFDocument* since it directs how the document should be parsed.
2. *swoogle:useSWLanguage* shows the Semantic Web languages used by an RDF document. Swoogle considers four meta-languages, namely "OWL", "DAML", "RDFS", and "RDF".
3. *swoogle:isOWLSpecies* shows the language species of an RDF document written in OWL. There are three possible species, namely "OWL-LITE", "OWL-DL", and "OWL-DL or FULL".
4. *wob:creator* shows the creator of the document. Its range is *foaf:Agent*.

Ontology level properties cover the annotation of an ontology.

1. *rdfs:label.*
2. *rdfs:comment.*
3. *owl:versionInfo*. It is OWL version of *daml:versionInfo*.

[3] It is a joint contribution by members in eBiquity group: Dr. Tim Finin, Dr. Anupam Joshi, Rong Pan, Pavan Reddivari, Vishal C Doshi and etc.

[4] Two RDF documents are identical unless they have the same md5hash; however, same md5hash does not guarantee two RDF documents are identical.

5.2.2 RDF graph Abstract

In many cases, users may only be interested in some properties of an RDF graph. Ontology engineers care the most relevant ontological terms to describe their conceptualisms. Publishers want to find the most popular classes and properties to encoding their knowledge. Readers want to find information about the same entity from multiple sources. Inference engines want to find appropriate ontologies to cover external reference of a given RDF graph. Hence, abstracting a given RDF graph, especially that serialized by RDF document, is highly desired to reduce search cost.

Existing approaches abstracts the serialized RDF graph. *Web search engines* treat RDF documents as text document and use bag-of-word approach to index RDF document by counting the term occurrence and frequency. *Swangle* [74] digest RDF document as a set of triples by generating Hash code for all seven possible wild-char queries. None of them take advantage of the semantics of URIrefs.

Out study reveals a series of models with different focuses on the information conveyed by an RDF graph.

1. **bag-of-word model.** This model focuses on the plain-text information in RDF graph, i.e. the local name of Resource nodes and the text of Literal nodes. Since the local name of Resource nodes are often compound words, we further split it in to atom words [5]. Hence an RDF graph is digested by a list of atom words. Currently, we treat the extracted words "as is" and leave IR stemming, word ordering for future work.

2. **bag-of-URIref model.** This model focuses on the named RDF nodes in RDF graph, i.e. the URIrefs of non-blank Resource node. Hence an RDF graph is digested by a list of URIrefs (without order). Note that we does not consider the URIrefs which are invalid according to RFC 2396 [4].

3. **triple model.** This model focuses on the RDF triples in RDF graph. Swangle digests a triple for all possible triple queries. We may further build a list these hash codes.

4. **ontological-term model.** This model focuses on ontological terms. It uses RDFS and OWL semantics to capture document-term relation, i.e. whether a class or property is defined, populated or referenced by RDF graphs.

5. **namespace model.** This model focuses on namespaces used and defined by RDF graph. It uses URI definition to capture namespaces of URIrefs in RDF graph.

6. **identity model.** This model focuses on the identity of defined classes, properties, and individuals. It uses RDFS and OWL semantics to digest non-local URIrefs of identities, and the value of *owl:reverseFunctionalProperty*.

These models share a common feature, an RDF graph is digested by one or several lists of terms/words. Therefore, RDF graph digest is converted into a conventional term document relation problem. Besides TFIDF methods, we can use *Bloom filter*

[5] Thanks Rong Pan contributing this idea.

[9], which has been proved as a space efficient tool for digesting keys in distributed network applications [31, 12]. With this method, an RDF graph can be digested by a bloom filter string regardless of the order of triples or RDF syntactical encoding. In this sense, Bloom filter works different from simply hashing RDF document.

In implementation, we should first estimate the size of keys n per RDF graph. Then propose appropriate number k of hash functions and size m of Bloom filter vector. As shown in [31], in order to guarantee low false positive rate, m should be at least twice larger than n and the hash function should be random enough.

5.2.3 Term Definition

An ontological term is defined by two types of information in RDF graph, namely term definition and class-property bond as shown in figure 5.1. Besides annotative properties, some *term definition* associates class/property with other resources via the relevant properties which imply *swoolge:definesClass* and *swoogle:definesProperty* (see section 5.2.4).

Class-property (C-P)) bond shows the properties that can be used to modify the instances of a given class. It is important in understanding class and composing class instances. C-P bond can be found in three situations:

- *ontological C-P bond* is captured by *rdfs:domain* in RDFS semantics, and by *owl:onProperty* in OWL semantics.
- *empirical C-P bond* is learned from class instances.

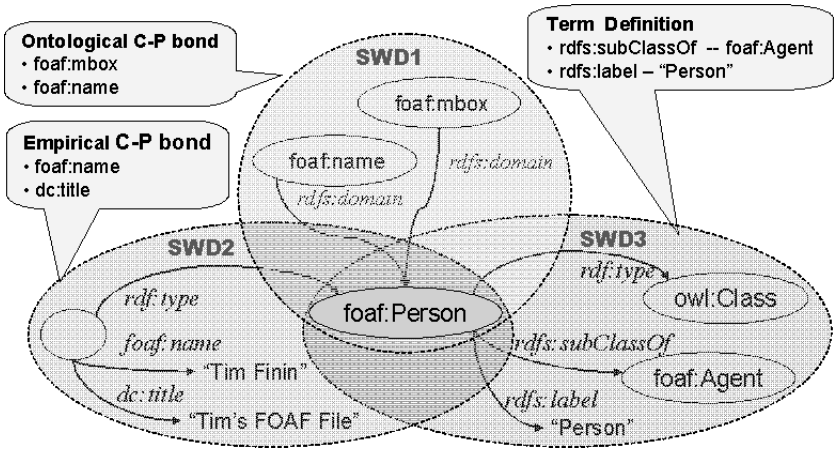

Fig. 5.1: Definition of class and class-property bond

5.2.4 Document and Resource Relation

We first consider the relation between RDF documents and ontological terms. An ontological term is either a named class or a named property. In an RDF graph, a node

is recognized as a *named class* iff. it is not an anonymous node and it is an instance of *rdfs:Class*; similarly, a node is a *named property* iff. it is not an anonymous node and it is an instance of *rdf:Property*. Swoogle considers two basic types of document-term relation, namely *swoogle:defines* and *swoogle:uses*.

1. **swoogle:definesClass.** If an RDF document D contains a triple whose predicate is one of the following: *rdf:type*[6], *rdfs:subClassOf*, *owl:complementOf*, *owl:disjointWith*, *owl:equivalentClass*, *owl:intersectionOf* ,*owl:oneOf*, *owl:unionOf*, the subject resource is called a defined class in D. It is sub-property of *swoogle:defines*.

2. **swoogle:definesProperty.** If an RDF document D contains a triple whose predicate is one of the following: *rdf:type* [7], *rdfs:subPropertyOf*, *owl:equivalentProperty*, *rdfs:domain*, *rdfs:range*, *owl:inverseOf*, the subject resource is called a defined property in D. It is sub-property of *swoogle:defines*.

3. **swoogle:populatesClass.** If an RDF document D contains a triple whose predicate is *rdf:type*, the object resource is called a populated class in D. It is sub-property of *swoogle:uses*.

4. **swoogle:populatesProperty.** All predicate resource in an RDF document D are called populated property in D. It is sub-property of *swoogle:uses*.

5. **swoogle:referClass.** If an RDF document D contains a triple whose predicate is one of the following: *rdfs:domain*, *rdfs:range*, *owl:allValuesFrom* , *owl:complementOf*, *owl:disjointWith*, *owl:equivalentClass*, *owl:intersectionOf*, *owl:someValuesFrom* , *owl:unionOf*, the object resource is called a referred class in D. It is sub-property of *swoogle:uses*.

6. **swoogle:referProperty.** If an RDF document D contains a triple whose predicate is one of the following: *rdfs:subPropertyOf*, *owl:equivalentProperty*, *owl:inverseOf*, *owl:onProperty*, the object resource is called a referred property in D. It is sub-property of *swoogle:uses*.

Another important relation is to determine the "official" ontology and "extension" ontology of a namespace. This is very important for inference engine for handling external reference.

1. **swoogle:officialOnto**. Given an RDF resource, the URL of its official ontology can only be i) the namespace of RDF resource; ii) the redirected URL of the namespace (e.g. `http://purl.org/dc/elements/1.1/` is redirected to `http://dublincore.org/2003/03/24/dces`); iii) the URL having the namespace as its absolute path (e.g. `http://xmlns.com/foaf/0.1/index.rdf` is the official ontology of `http://xmlns.com/foaf/0.1/`).

2. **swoogle:extensionOnto**. Given RDF resource R, all ontologies that defined R but were not the official ontologies are called extension ontology of R.

[6] plus the object is subclass of rdfs:Class.
[7] plus the object is subclass of rdf:Property.

5.2.5 Inter-Document Relation

In semantic web, RDF documents are related by properties from meta-ontologies to facilitate users finding term definition and resolving the external term reference.

1. In **RDF semantics**, documents are associated by *rdfs:seeAlso* and *rdfs:isDefinedBy*. It is notable that these properties are not necessarily link to RDF documents; hence RDF validation is needed to assure such document relation.

2. In **OWL semantics**, ontologies are associated by sub-properties of *ow:OntologyProperty* including *owl:imports*, *owl:priorVersion* , *owl:backwardCompatibleWith*, and *owl:incompatibleWith*.

In previous Swoogle practice, we did generalize inter-term relation to inter-document relation[25]. However, the generalized inter-document relation largely depends on co-occurrences of terms. In the case that a class c is referenced by M RDF documents and defined by N ontologies, there will be $M \times N$ inter-document relations. Such approach has scaling limitations.

5.3 Searching Terms and Documents

The web-scale semantic web differs from database system since users will obtain data from the Web; it differs form IR system since it organize data with RDF graph; and it also differs from the Web since RDF documents are not well connected by hyperlinks. Hence semantic web data access involves searching the Web and searching RDF graph. The semantic web digest information constitutes the basis of such search.

5.3.1 Searching Ontological Terms

Swoogle *Ontology Dictionary* collects definition/usage information about terms (classes and properties) defined in the Semantic Web. Currently, ontology dictionary provides two search services for ontological terms: *Term Lookup* and *Alphabetical Term Index*.

Swoogle's *Term Lookup*, as shown in figure 5.2, searches ontological terms that meet the provided constraints such as *local-name*, *namespace*, and *whole URI*. In addition, it supports two filters: i)*term family filter*, i.e., which meta ontology is used to define that term, and ii) *term type filter*, i.e., whether the term is defined as class and/or property. Since URIrefs are case-sensitive, *http://example.com#person* is different from *http://example.com#Person*. Swoogle also allows user specify keyword matching semantics, i.e., exact match, prefix match, suffix match, and fuzzy match (substring match).

Swoogle's *Alphabetical Term Index*, as shown in figure 5.3, lets users browse ontological terms by prefix in ascending order. It has two views: the prefix-view is to the left and the matched-term-list view is to the right. In the prefix-view, each prefix is followed by the amount to terms using that prefix. In the matched-term-list, all terms matching current prefix are listed.

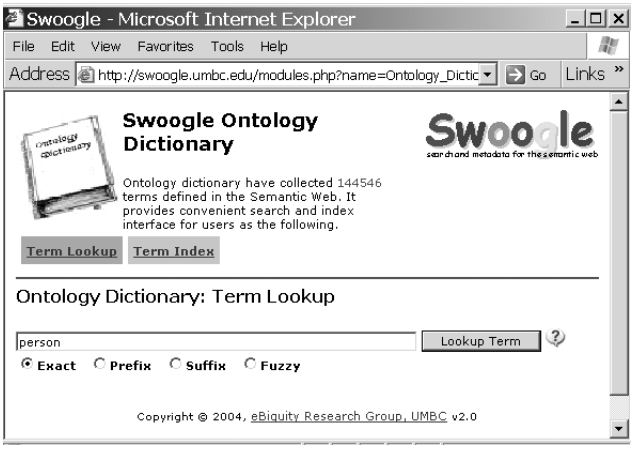

Fig. 5.2: Swoogle term lookup interface

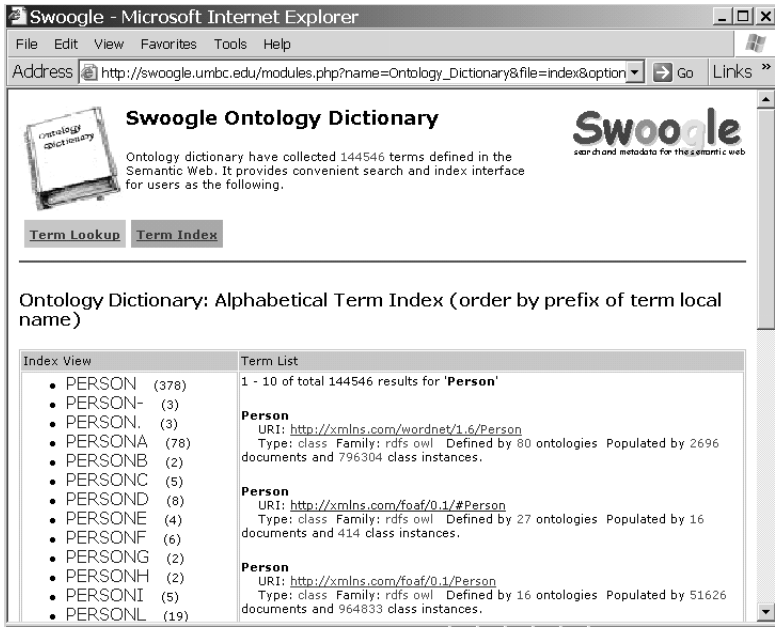

Fig. 5.3: Swoogle alphabetical term index interface

5.3.2 Searching Ontologies

Swoogle's *Document Search* (see figure 5.4)[8] helps users to locate relevant ontologies with constraints on: the URL, file extension and syntactical encoding of ontology. It also allows users to specify constraints on the terms defined or used by ontology.

[8] Document search is developed by Rong Ran.

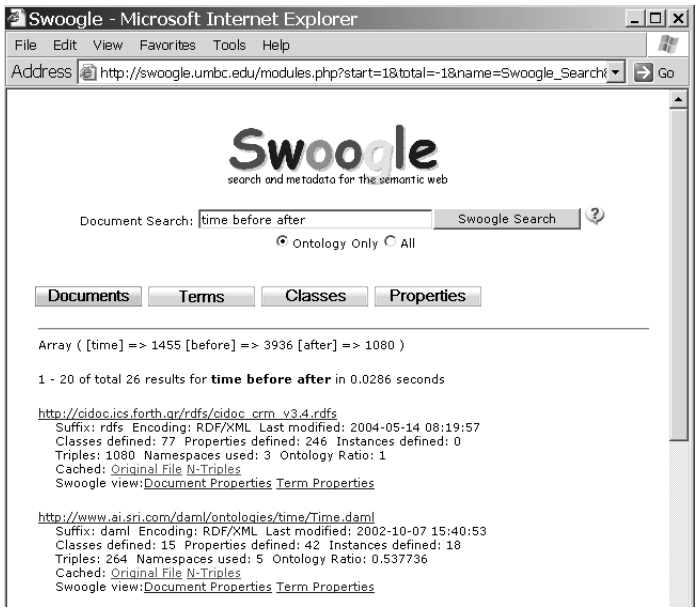

Fig. 5.4: Swoogle document search interface

5.3.3 Navigating Semantic Web

Besides search, Swoogle also provides other canned browse mechanisms for users to browse the Semantic Web: a user may start from an ontological terms or an ontology and then navigate other terms or ontologies (see figure 5.5). Note *owl:Ontology* is a subclass of *wob:RDFDocument*.

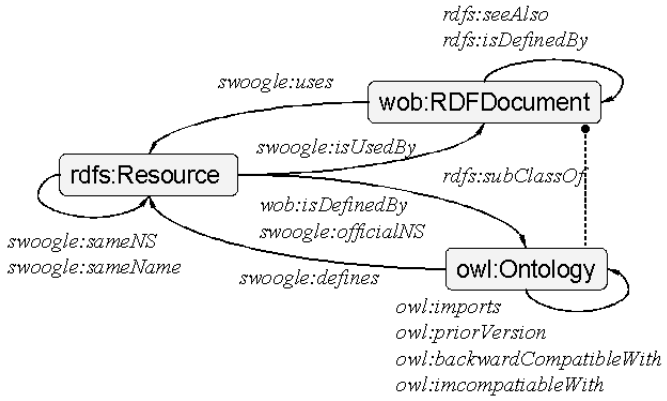

Fig. 5.5: Navigation paths in the Semantic Web

Given an ontological term, a user may navigate via the following paths:

- *wob:isDefinedBy* leads to term definition provenance, i.e. search all ontologies that defines the given term.
- *wob:sameNamesapce* leads to terms sharing namespace with the given term.
- *wob:sameName* leads to terms sharing name with the given term.
- *swoogle:uses* leads to RDF documents that uses the given term.

Given an ontology, a user may navigate through the following paths:

- swoogle:defines leads to terms defined by the ontology
- swoogle:uses leads to terms used by the ontology
- owl:imports,owl:priorVersion *owl:backwardCompatibleWith*, and *owl:imcompatiableWith* lead to relevant ontologies.
- *rdfs:seeAlso* and *rdfs:isDefinedBy* lead to related documents.

6. PRELIMIARY WORK: EVALUATING SEMANTIC WEB QUALITY

6.1 Identifying Data Quality Dimensions

Identifying the dimensions of data quality is essentially modeling information imperfectness, and this process may never be complete due to our inherent ignorance and limited evaluation capabilities. Based on our review of imperfectness representation in appendix B, we proposed a mapping between dimensions of data product quality (based on [101]) and imperfectness (based on Smithson's imperfectness taxonomy[97]) in table 6.1.

Tab. 6.1: Mapping data quality dimensions to imperfectness

dimension	cited	ontological dimension	imperfectness
accuracy	25	garbled mapping to wrong state	inaccuracy, vagueness
reliability	22	correct	probability
timeliness	19	wrong, meaningless, ambiguous state	ambiguity
relevance	16	n/a	irrelevance
completeness	15	full mapping	absence

In WOB conceptualism, we need to consider quality issues in different granularities, namely resource, RDF document and RDF graph.

6.1.1 Quality of RDF Resource

An RDF resource is used to refer a thing in real world. Currently, we consider two quality dimensions of a special type of RDF resource –*ontological term*: i) relative term vagueness, which shows that one RDF resource has more vague semantics than the other; and ii) term importance, which shows a term's impact in semantic web data access.

6.1.2 Quality of RDF Document

An RDF document is used to serialize an RDF graph in the Web. Currently, we consider one quality dimension of a subclass of RDF document – *ontology*: document importance, which shows an ontology's impact in semantic web data access. Note we will discuss the quality serialized RDF graph next.

6.1.3 Quality of RDF Graph

RDF graph is used to describe the author's interpretation of the real world. Currently, we consider two categories of quality dimensions: *context based* and *trust and provenance based*.

Content Based Quality Dimensions

Content based quality dimensions include both *direct RDF graph quality* and *relative RDF graph quality*.

Direct quality of RDF graph. An RDF graph can be directly evaluated by semantic web reasoner and complex network analysis tools on the following aspects:

- **Semantic consistency** shows whether an RDF graph is consistent according certain semantics. We assume the RDF graph is valid, and then check its consistency using either RDFS or OWL semantics.
- **Definition closeness** shows whether an RDF graph has undefined ontological terms, i.e. each ontological term T should be defined by at least one triple in form of $(T, rdf : type, \_y)$.
- **Graph structure properties** include connectivity, degree distribution, cluster coefficient and etc.

Relative quality of RDF graph. In many cases, quality is rather a subjective judgment and requires RDF graph comparison. Let GT be the target RDF graph and GU be the reference RDF graph.

- **rel-completeness** How much does GT entail GU? Follow this line is "RDF graph Diff" problem[1].
- **rel-consistency** If we merged GT and GU, will the merged RDF graph be consistent?

Trust and Provenance based Quality Dimensions

Besides content analysis, a user may also evaluate an RDF graph using context information. Provenance relates RDF graph with *wob:RDFDocument* and *foaf:Agent*. Trust explicitly hypothesizes how good agents can interpret the real world by authoring RDF graph.

6.2 Ranking RDF documents and RDF Resources

We rank RDF documents and resources by their importance in Semantic web data access. We constructed a list of ranking problems under two concerns: i) how much semantics we will use; and ii) what will be ranked.

[1] http://www.w3.org/DesignIssues/Diff

6.2.1 Ranking RDF Resources Using Node-edge Interpretation

The problem is to rank RDF nodes in an RDF graph. We treat an RDF graph as a directed graph, each edge of which is associated with a name (i.e. URIref). The corresponding navigation model is shown in figure 6.1, and users simply navigate from one RDF node to another via named edges.

Fig. 6.1: RDF graph navigation model 1

Let $G(V, E)$ be an RDF graph, V is the set of RDF nodes, and E is the set of triples, which correspond to the directed edges between RDF nodes. Since all edges are named with URIrefs, let NE be the set of edge names. The semantics associated with edges help information consumers to browse information selectively, and thus raise the issue of biased surfing, which is caused by assigning weights to edges according to the importance of their names.

Based on the intuition that information consumers like to follow familiar/important edges, we use edge name frequency of RDF graph (either supplied by the user or the given RDF graph) to assign edge weight. We compute rank by a variation of PageRank algorithm: let $R(x)$ be the rank of an RDF node, $w(y)$ be the weight of a edge with name y, $R(x)$ is computed by equation 6.1.

$$R(o) = d + (1 - d) \sum_{(s,p,o) \in E} R(s)w(p) \tag{6.1}$$

6.2.2 Ranking RDF Resources Using RDFS Semantics

The problem is to rank RDF nodes in an RDF graph. But we take RDFS semantics in count. The corresponding navigation model (see figure 6.2) assumes that users understand RDFS vocabulary and thus treat class, property, individual differently. We should note triple-property relation, i.e., some links to *property* are derived from triples: a user may visit the definition of the predicate when reading the subject of a triple.

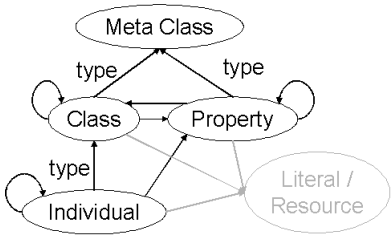

Fig. 6.2: RDF graph navigation model 2

Figure 6.3 can be used to explain this navigation model. A user starts from `http://foo.com/ex.owl`, and then he may follow any of the following links: i) go to *wob:RDFDocument* via *rdf:type*; ii) go to *wob:sourceDocument* via the triple (`http://foo.com/ex.owl`, wob:sourceDocument, *a2*) ; and iii) go to *a2* via *wob:sourceDocument*.

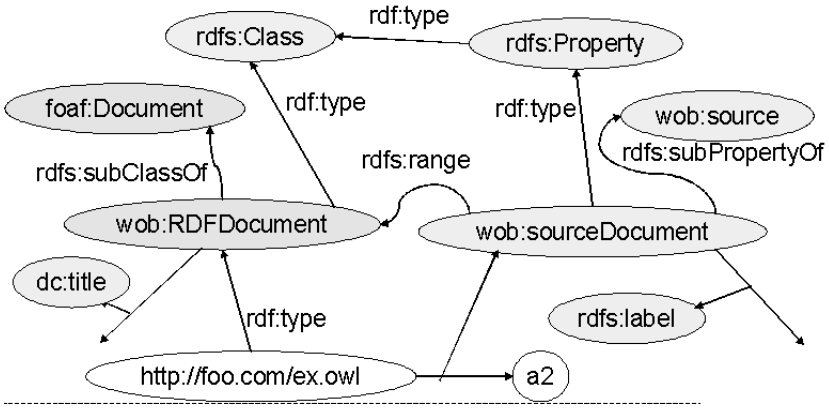

Fig. 6.3: An example RDF graph

This navigation model focuses on ontological terms and class instance URIrefs and skips a lot of resources and all *rdf:Literal* in this navigation model. In addition only properties in RDFS ontology plus the triple-property relation will be used in navigation.

The biased surfing model is still effective, but the edge weight may be assigned by either edge-name frequency or by empirical constants [2].

6.2.3 Ranking RDF Resources/Documents Using WOB Semantics

Here we may rank RDF resources and RDF documents together using WOB metadata. The corresponding navigation model (see figure 6.4) assumes that users use Swoogle to navigate the Semantic Web. The ranking algorithm is for future work.

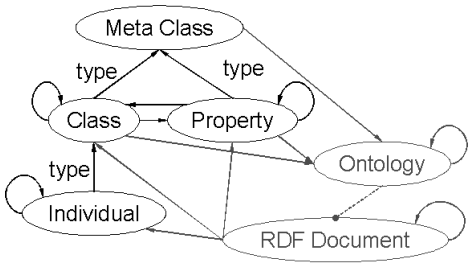

Fig. 6.4: RDF graph navigation model 3

[2] we used empirical constants in swoogle Ontology Rank.

6.3 Evaluating RDF graph trustworthiness

Given a set of statements extracted from the Semantic Web, how much should we trust in the model they describe? This can be viewed as a problem central to document analysis in which not all information sources are trusted at the same degree and it has obviously important applications in the homeland security domain. For example, the "CIA World Fact Book" and "NASDAQ" are highly trusted while "The National Enquirer" is somewhat less trusted. The trustworthiness of a semantic association from "Mr. X" to "Osama Bin Laden" (as mentioned in introduction section) depends on the analyst's trust in the agents in the provenance chains. The situation may be more complex when "Agent K" and "The National Enquirer" have conflicting beliefs over the statement "Organization B invests Company A".

We model the solution to these two problems using the following notation: $S = \{s_1, s_2, ..s_n\}$ be as a collection of n RDFS statements, Z be the analyst agent, $T(x, y)$ be the trust value from agent x to agent y, $TH(x)$ be the set of highly trusted agents by x, $C(x, s)$ be the trustworthiness of a collection of statements s is true according to agent x, $Pro(s)$ be the collection of agents that are the provenance of a statement s, $AB(s)$ be the collection of agents who has belief over a statement s.

First we examine a simple situation in which statements are independent, semantically consistent, and fully believed by their provenance agents. We build a Bayes probabilistic model of knowledge sources based on "Noise-Or" theory [83]. Equation 6.2 assumes that provenance agents are independent and that their knowledge accuracy is correctly captured as trust. Therefore, we use "Noise-OR" method to derive the probability that statement s_i is true to agent A given A's trust to the provenance agents $Pro(s_i)$. This followed by a multiplication which aggregates the overall confidence since all statements are independent.

$$C(Z, S) = \prod_{s_i \in S} \left(1 - \prod_{x \in Pro(s_i)} (1 - T(Z, x)) \right) \tag{6.2}$$

Using this technique, if analyst Z's trust in NASDAQ is $T(Z, NASDAQ) = 0.99$, Z's trust in "The National Enquirer" is $T(Z, FOO) = 0.5$, Z's trust in "Agent K"is $T(Z, K) = 0.6$, Z's trust in "CIA Agent W" is $T(Z, W) = 0.8$, then $C(Z, S0)$, where S0 refers to the semantic path from "Mr.X" to "Osama Bin Laden"(as mentioned in introduction section), is $0.99 \times (1 - (1 - 0.5)(1 - 0.6)) \times 0.8 \cong 0.63$. This path is much more trustworthy than the cases that only one of "Agent K" and "The National Enquirer" is the provenance agent.

The second situation is more complicated since (i) inconsistent statements may be detected according to ontological semantics (e.g. a person's name can not have two different values) and (ii) more agents beside the provenance agents may assert beliefs through statement reference. A straightforward approach is consensus model which is based on the intuition that trusted peers' opinions are the only sources of reliable information. Equation 6.3 averages the discounted belief confidence from trusted agents.

$$C(Z, S) = \prod_{s_i \in S} \left(\sum_{x \in AB(s_i) \cap TH(Z)} \frac{T(Z, x) * \hat{B}(x, s_i)}{|AB(s_i) \cap TH(Z)|} \right) \tag{6.3}$$

where $\hat{B}$ is the confidence value of belief/disbelief (note that we change the sign of belief confidence for disbelief). Assume that the statement $s1$ " Organization B invests Company A" is believed by "The National Enquirer" but disbelieved by "Agent K" and both have belief confidence 1. For analyst Z, $C(Z, \{s1\})$ is $(-0.9 + 0.5)/2 = -0.2$ when $T(Z, K) = 0.9$ and $T(Z, FOO) = 0.5$, and is $(-0.5 + 0.5)/2 = 0$ when $T(Z, K) = 0.5$ and $T(Z, FOO) = 0.5$. In both case, s1 is weakly believed by Z indicating that more field investigation is needed. In addition, the absolute value of derived confidence shows that the first case should be investigated first.

6.4 Trust based navigation and knowledge expansion

Our work uses a trust network to navigate the Semantic Web and incrementally incorporates external knowledge sources. The trust and provenance based heuristic is described below. Its input includes: agent A who conduct the inference, the $query$ involved in the inference, A's trust network TN_A, a customizable trust threshold α and a social distance limit β.

Inference(A**,** $query$**,** TN_A**,** α**,** β**):**
1. distance=0, KB={}, Agents ={}
2. if (distance $> \beta$) return;
3. newAgents = find_agents(TN_A, A, Agents, α, distance)
4. if (newAgents is empty) then return fail
5. Agents = Agents $\cup$ newAgents
6. KB = merge_knowledge(TN_A, A, newAgents, KB)
7. doInference(KB, $query$)
8. if ($query$ is answered) then return with result
9. else distance++ and go to step 2

We leave the details of how different types of trust (namely referral trust and domain trust) are used to implement "find_agents" and "merge_knowledge" to our previous work [27, 26]. The "doInference" function will run a conventional inference. According to the algorithm, inference is run iteratively when expanding social distance, and has three normal terminating conditions: an answer is found, no more trust agents can be found, or the social distance limit is reached. Therefore, the space complexity, which is the union of trusted agents' knowledge within certain social distance, is bounded by social distance and trust threshold. In addition, "doInference" will run at most β times.

The performance of trust and provenance based heuristic depends on the following conditions: (i) relevant knowledge always comes from the same sources and shares provenance; (ii) trusted agents are likely to be good information sources to the trustor, i.e. having useful and consistent knowledge; and (iii) the customized trust network is correctly derived (in practice, the trust network may be also dynamically derived and evolved to reach better accuracy [27]).

With such heuristics, semantic association discovery, which can be abstracted as finding sub-graphs in RDF graph, will first use knowledge from the most trusted in-

formation sources like “Agent K” and then expand to less reliable sources like “The National Enquirer”. The trust threshold assures the reliability of analytical results. When too many sources are trusted, domain filters may play a good role in reducing complexity (without touching sources not trusted).

7. CONTRIBUTIONS TO COMPUTER SCIENCE

The contributions of this dissertation are the following:

- WOB is one of the first attempts that make the Semantic Web self-descriptive using OWL semantics. The WOB conceptualism models the Semantic Web and its context, and the WOB ontology helps representing the metadata of the Semantic Web in RDF graph. Rather than bringing more annotative information, WOB ontology use OWL semantics to make semantic web metadata suitable for inference.
- Swoogle is one of the first data access services that digest and search the web-scale semantic web. It is highlighted by the following features:
  - Keyword based adaptive semantic web discovery agent.
  - RDF graph abstract which summarize RDF for search.
  - Semantic web navigation models.
- We are among the pioneers who rank the Semantic Web. We proposed a list of navigation model for ranking ontological terms and RDF documents in the Semantic Web. *Swoogle Statistics* is a byproduct of our research, and it quantifies the web-scale semantic web with statistics.
- We are among the pioneers who evaluate RDF graph's trustworthiness and contribute to semantic web trust layer. We proposed a list of trust evaluation mechanism according the different background knowledge. We also developed a trust and provenance based navigation model that supports P2P style knowledge expansion.

APPENDIX

A. THE GROWING PRACTICE OF THE SEMANTIC WEB

The past year has seen a dramatic growth of the Semantic Web practice. Swoogle [25] has reported over 300,000 RDF documents and amongst 4,000 ontologies written in RDF or OWL on the Web.

The increase of RDF data is accelerated by the joint force of industry and academic research. Industry adoption of RDF greatly increases the amount of online RDF data. As shown in figure A.1, news feeds using RDF Site Summary(RSS) [1] and Personal profile using Friend Of A Friend ontology (FOAF) [2] are the main streams of current RDF data world. Though such data is automatic generated and used as XML, it does best retain the semantics and offers better opportunities for third party users.). Another source of RDF data is web services which provide structured information based on their underlining database. Some already use RDF to encode the output[108, 25]; the others with XML output can be translated into RDF easily (e.g. Google Web API[3] and Amazon web Services[4]). More such information is expected with the advance of automatic document annotation, entity extraction and link detection tools such as Semagix Freedom[6] and IBM's tools[24].

(generated at: Mon, 27 Dec 2004 17:30:13 -0500)

There are 4561 namespace used by 326256 SWDs

rank	namespace URI	SWD used	SWD defined
1	http://www.w3.org/1999/02/22-rdf-syntax-ns#	289252 (97.8%)	191 (0.1%)
2	http://purl.org/dc/elements/1.1/	210759 (71.3%)	197 (0.1%)
3	http://purl.org/rss/1.0/	180954 (61.2%)	12 (0.0%)
4	http://webns.net/mvcb/	96412 (32.6%)	0 (0.0%)
5	http://xmlns.com/foaf/0.1/	61896 (20.9%)	76 (0.0%)
6	http://www.w3.org/2000/01/rdf-schema#	50677 (17.1%)	183 (0.1%)

Fig. A.1: Swoogle's six best namespace in the Semantic Web

[1] http://purl.org/rss/1.0/
[2] http://xmlns.com/foaf/0.1/
[3] http://www.google.com/apis/
[4] http://www.amazon.com/gp/aws/landing.html

We also observed the growth of ontology, though not in large amount. As shown in figure A.1, besides the meta-ontologies (RDF, RDFS, OWL), there are huge amount of ontologies covering various domain, e.g. digital library metadata, newsfeed, personal information, software configuration, bibliography, date/time and copyright. Some ontologies are well populated but much more only have a few or zero population. In addition, many ontolgoies are dialects of the same domain, e.g. person has been defined under more than 150 namespaces.

Accompany with the growth of information, more semantic web tools are released to promote the usage and visibility of the semantic web. Tools include editors (e.g. IsaViz, Protg, Swoope, Orient), reasoners e.g. cwm,Euler,Jena,f-owl, and etc, parsers and validators(e.g. jena, bbn), storage (sesame, jena, kowali), portals(semanticwebcentral, swoogle, ontaria) and more[5].

[5] more tools may be found at http://www.w3.org/2001/sw/WebOnt/impls

B. REVIEW OF IMPERFECTNESS FORMALISMS

Imperfectness exists because of our ignorance of the world. Studies on imperfectness may date back to Bayes' time (18th centry); however, the first concrete alternative uncertainty framwork, fuzzy set theory, come into being in 1960s [97]. The ten years between 1988 to 1998 appears to be a fruitful with many published books [97, 68, 65, 52, 77], proceedings [53] and surveys [28, 95, 96, 81, 82, 58].

The very first issue is to understand imperfectness itself[1]. As phenomenon, imperfectness is always explained its sources through typologies. Smithoson [97] suggested a taxonomy of ignorance, which differentiated uncertainty from absence and pointed three sources of uncertainty (vagueness, probability, and ambiguity) . Smets [95] suggested three varieties of ignorance, namely incompleteness, imprecision and uncertainty. Klir and Yuan [65] shows two sources of uncertainty, namely fuzziness and ambiguity. Parsons [81] summarized five sources of imperfectness, namely uncertainty, imprecision, incompleteness, inconsistency and ignorance. Parsons and Hunter [82] further stress the distiction between uncertainty and absence in Smithoson's taxomony. A different typology come from cognitive perspective, where uncertainty of phenonmenon should be deferentiated from the uncertainty of its representation. Regan et al., [87] differentiated Linguistic (verbal) uncertainty from epistemic uncertainty and stressed the influence of social context.

The second issue is to model imprefectness in terms of representation and computation. According to Parsons and Hunter [82], uncertainty and absence results in the "numerical camp" and "symbolic camp" repectively. "numerical camp" adopts quantitative meaures to handle uncertainty. Representative approaches under this camp are: probabilistic theory, possibility theory, and evidence (Demperster-Shafer) theory. "symbolic camp" address absence using logic methods. Representative approaches under this camp are: default logic, argumentation, truth maintenance system, and answer set theory. A brief survey is done by Parsons and Hunter [82] and more details may be obtained from [97, 68]. In addition to those two camps, Druzdzel [28] argued that verbal phrases are better alternatives to numbers in describing uncertainty in many real world cases.

The third issue is to use and evaluate imperfectness formalisms in information systems, such as database, knowledge base, information retrieval system, data mining. The usage normally consists of three steps: i) identify the types of uncertainty in that system and build a mapping (between number and verbal phrase) when necessary, ii) measure and represent uncertainty of each atoms; iii) define uncertainty computation

[1] We should note that this is rather a inter-dicipline issue including computational, epistemic, cognitive, social and many more aspects

w.r.t. the information systems' native operation.

WE NEED TO PROVIDE A TABLE TO COMPARE RELATED WORK.

C. REVIEW OF COMPUTATIONAL TRUST

Trust is a cross discipline research topic (e.g. philosophy, sociology, psychology and computer science)[34, 72, 78]. Our enormous real life "trust" related experiences resulted in various and never-ending definition of trust [34, 76, 35]. This dissertation focuses on the computational trust, which is pioneered by [72] and featured by the following threads [1]:

Security Traditional security adopts the authenticity and integrity semantics of trust. "The purpose of modeling trust must therefore be to model how a human observer would asses the security of a system or the honesty of an agent"[59]. the word "web of trust" was first used by PGP [107] ; however, it and X.509 are well known protocols in this thread. [14]

Distributed trust management Trust management, coined by Blaze [8], use authorization policy to capture operational semantics of trust. It focuses on "collecting the information required to make a trust relationship decision , evaluating the criteria related to the trust relationship as well as monitoring and reevaluating existing trust relationships"[8]. Trust management is different to traditional security protocol based approach [38, 84], and it proliferates a series of work including PolicyMaker[8], KeyNote[7], Referee[18], Delegation logic [69], Rei (Semantic Web policy language) [32, 61], Sultan [39].

Trust in multi-agent system In multi-agent system, trust is characterized as an intermediate hypothesis about agent' behavior involving several processes: i) learning trust from past experience; ii) formalizing trust into decisional variables; iii) using trust to decide the next action. Such research can be viewed as an extension of multi-agent reinforcement learning [60, 51] since it is highlighted by the formalization of trust and social control. Trust formalization offers better explanation to agent decision, and social control [86] adopts recommendation [105] and thus enables collaborative decision. [1, 106] assume some global information to be maintained at each individual agent or relying on a centralized server. [2, 104] has been proposed as a mechanism for building trust in P2P electronic communities.

Reputation System Reputation is a relevant concept to trust. However, it is different to trust [78]

[1] This classification intends to show the board interests in computational trust, and it is neither comprehensive and disjunctive. More classifications can be found in other surveys [38, 85]

Trust network trust network is the advanced stage of trust in multi-agent system since it focus on the network of referral trust. Inspired by Google's PageRank, researchers [37, 62, 89, 88], explicitly refer trust network and approximate *global trust* by propagating local trust through the trust network. However, the meaning and valuation of referral trust in these graph theoretic approaches is rather heuristic and not well justified.

In future work , we will review existing trust models with three aspects: *representation*, *computation* and *evaluation*.

D. WOB CORE ONTOLOGY

The WOB core ontology depends on two existing ontology, namely Dublin Core Element ontology (v1.1) and FOAF ontology. In order to keep these ontology in manageable scale,i.e. keep them in OWL Lite or DL level, we selcted some terms from these two ontologies. This work results in three ontologies.

- http://daml.umbc.edu/ontologies/webofbelief/1.1/dc.owl
- http://daml.umbc.edu/ontologies/webofbelief/1.1/foaf.owl
- http://daml.umbc.edu/ontologies/webofbelief/1.1/core.owl

The proposed web core ontology (v1.1) is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
     <!ENTITY rdf  "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
     <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
     <!ENTITY xsd  "http://www.w3.org/2001/XMLSchema#" >
     <!ENTITY owl  "http://www.w3.org/2002/07/owl#" >
     <!ENTITY dc  "http://purl.org/dc/elements/1.1/" >
     <!ENTITY foaf   "http://xmlns.com/foaf/0.1/">
     <!ENTITY wob   "http://daml.umbc.edu/ontologies/webofbelief/1.1/core.owl#" >
   ]>


<rdf:RDF
  xmlns      = "&wob;"
  xmlns:wob  = "&wob;"
  xml:base      = "http://daml.umbc.edu/ontologies/webofbelief/1.1/core.owl"
  xmlns:dc   = "&dc;"
  xmlns:foaf    = "&foaf;"
  xmlns:rdf  = "&rdf;"
  xmlns:rdfs  = "&rdfs;"
  xmlns:owl  = "&owl;"
>

<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://daml.umbc.edu/ontologies/webofbelief/1.1/foaf.owl"/>
  <owl:imports rdf:resource="http://daml.umbc.edu/ontologies/webofbelief/1.1/dc.owl"/>
  <rdfs:label xml:lang="en-US">Web of Belief Ontology Core Elements</rdfs:label>
  <rdfs:comment xml:lang="en-US">
This file specifies the core elements of WOB ontology.
created by Li Ding. http://www.csee.umbc.edu/~dingli1/foaf.rdf.
  </rdfs:comment>
  <owl:versionInfo xml:lang="en-US">
21 December 2004, revised $Date: 3:12 PM 12/21/2004$
  </owl:versionInfo>
  <owl:priorVersion>
<owl:Ontology rdf:about= "http://daml.umbc.edu/ontologies/webofbelief/1.0/Wob.owl"/>
  </owl:priorVersion>
  <dc:language xml:lang="en-US">English</dc:language>
```

```
  <dc:creator>
<foaf:Person>
<foaf:name>Li Ding</foaf:name>
<foaf:mbox_sha1sum>ba7653bf90ad13f74795696a659f89b741396916</foaf:mbox_sha1sum>
<rdfs:seeAlso rdf:resource="http://www.csee.umbc.edu/~dingli1/foaf.rdf"/>
</foaf:Person>
  </dc:creator>
</owl:Ontology>


<!-- RDF Document-->
<owl:Class rdf:ID="RDFDocument">
<rdfs:subClassOf rdf:resource="&foaf;Document"/>
   <rdfs:comment xml:lang="en-US">
The abstract statement reference. It can reference any set of staments
with location constraints.
rdfs:Statement is a subclass of it.
</rdfs:comment>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#useRDFsyntax"/>
<owl:cardinality> 1 </owl:cardinality>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="RDFSyntax">
<rdfs:label xml:lang="en-US">RDF Syntax</rdfs:label>
   <rdfs:comment xml:lang="en-US"> The RDF document syntax </rdfs:comment>
</owl:Class>

<RDFSyntax rdf:ID ="RDF-XML">
<rdfs:label xml:lang="en-US">RDF/XML</rdfs:label>
</RDFSyntax>
<RDFSyntax rdf:ID ="N3">
<rdfs:label xml:lang="en-US">N3</rdfs:label>
</RDFSyntax>
<RDFSyntax rdf:ID ="NT">
<rdfs:label xml:lang="en-US">N-Triple</rdfs:label>
</RDFSyntax>

<owl:ObjectProperty rdf:ID="useRDFsyntax">
  <rdfs:domain rdf:resource="#RDFDocument"/>
  <rdfs:range rdf:resource= "#RDFSyntax"/>
  <rdfs:label xml:lang="en-US">RDF syntax</rdfs:label>
  <rdfs:comment xml:lang="en-US">
The syntax choice of RDF document.
  </rdfs:comment>
</owl:ObjectProperty>

<!--  RDF graph reference -->
<owl:Class rdf:ID="RDFGraphRef">
   <rdfs:comment>
The abstract RDF graph reference. It can reference any set of staments
with location constraints.
rdfs:Statement is a subclass of it.
</rdfs:comment>
</owl:Class>

<!--  provenance -->
<owl:ObjectProperty rdf:ID="source">
  <rdfs:comment>
 the source of information: i) an RDF grpah, ii) a resource, and iii)an RDF document.
  </rdfs:comment>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="sourceDocument">
  <rdfs:subPropertyOf rdf:resource="#source"/>
```

```
  <rdfs:range rdf:resource= "&foaf;Document"/>
  <rdfs:comment>
 where-provenance:  the document source
 It is the inference version of dc:source.
  </rdfs:comment>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="creator">
  <rdfs:subPropertyOf rdf:resource="#source"/>
  <rdfs:range rdf:resource= "&foaf;Agent"/>
  <rdfs:comment>
 whom-provenance: the agent(person or website) who created this information
 It is the inference version of dc:creator.
  </rdfs:comment>
</owl:ObjectProperty>


<owl:ObjectProperty rdf:ID="isDefinedBy">
  <rdfs:subPropertyOf rdf:resource="#source"/>
  <rdfs:range rdf:resource= "#RDFDocument"/>
  <rdfs:comment>
 definition-provenance: the RDF document that defined/declared this resource
 It is the inference version of rdfs:isDefinedBy.
  </rdfs:comment>
</owl:ObjectProperty>

<!-- Abstract Association -->
<owl:Class rdf:ID="Association">
   <rdfs:label>Association</rdfs:label>
   <rdfs:comment>
The abstract association
</rdfs:comment>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#connective"/>
<owl:cardinality> 1 </owl:cardinality>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:ObjectProperty rdf:ID="connective">
  <rdfs:domain rdf:resource="#Association"/>
  <rdfs:comment>
 shows the connective of the Association
  </rdfs:comment>
</owl:ObjectProperty>

</rdf:RDF>
```

BIBLIOGRAPHY

[1] Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *HICSS*, 2000.

[2] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In *CIKM*, pages 310–317, 2001.

[3] Alvaro Arenas, Brian Matthews, Michael Wilson, and Jan Grant. Vocabularies and architecture for implementing trust in the semantic web. `http://www.w3.org/2001/sw/Europe/reports/trust/11.2/d11.2_trust_vocabul%aries.html`, 2004.

[4] T. Berners-Lee, R. Fielding, and L. Masinter. Rfc 2396 - uniform resource identifiers (uri): Generic syntax. `http://www.faqs.org/rfcs/rfc2396.html`, 1998.

[5] Tim Berners-Lee and Dan Connolly. Delta: an ontology for the distribution of differences between rdf graphs. `http://www.w3.org/DesignIssues/Diff`, 2004.

[6] B.Hammond, A. Sheth, and K. Kochut. Semantic enhancement engine: A modular document enhancement platform for semantic applications over heterogeneous content. In *Real World Semantic Web Applications*. IOS press, 2002.

[7] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. RFC 2704: The keynote trust-management system version 2, 1999.

[8] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173, 1996.

[9] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[10] Brian E. Brewington and George Cybenko. How dynamic is the Web? *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):257–276, 2000.

[11] Dan Brickley. Scutter spec. `http://rdfweb.org/topic/ScutterSpec`.

[12] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. In *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing*, pages 636–646, 2002.

[13] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and where: A characterization of data provenance. In *International Conference on Database Theory (ICDT)*, pages 316–330, 2001.

[14] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. In *SOSP '89: Proceedings of the twelfth ACM symposium on Operating systems principles*, pages 1–13. ACM Press, 1989.

[15] Jeremy Carroll and Christian Bizer. The semantic web trust layer. `http://www.wiwiss.fu-berlin.de/suhl/bizer/pub/carrollbizer-trust-www200%4-devday.pdf`, 2004.

[16] Jeremy J. Carroll, Christian Bizer, Patrick Hayes, and Patrick Stickler. Named graphs, provenance and trust. Technical Report HPL-2004-57, HP Lab, May 2004.

[17] Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, MA, August 2004.

[18] Yang-Hua Chu, Joan Feigenbaum, Brian LaMacchia, Paul Resnick, and Matrin Strauss. Referee: Trust management for web applications. *World Wide Web Journal*, 2(3):127–139, 1997.

[19] Yingwei Cui, Jennifer Widom, and Janet L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Trans. on Database Systems*, 25(2):179–227, June 2000.

[20] Paulo Pinheiro da Silva, Deborah L. McGuinness, and Richard Fikes. A proof markup language for semantic web services. Technical Report KSL-04-01, Stanford, 2004.

[21] Paulo Pinheiro da Silva, Deborah L. McGuinness, and Rob McCool. Knowledge provenance infrastructure. *Data Engineering Bulletin*, 26(4):26–32, Dec 2003.

[22] John Davies, Richard Weeks, and Uwe Krohn. Quizrdf: search technology for thesemantic web. In *WWW2002 workshop on RDF and Semantic Web Applications, 11th International WWW Conference (WWW11)*, 2002.

[23] Stefan Decker, Michael Erdmann, Dieter Fensel, and Rudi Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In *DS-8*, pages 351–369, 1999.

[24] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, , and Jason Y. Zien. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *The Twelfth International World Wide Web Conference (WWW2003)*, 2003.

[25] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal C Doshi, , and Joel Sachs. Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, 2004.

[26] Li Ding, Pranam Kolari, Shashidhara Ganjugunte, Tim Finin, , and Anupam Joshi. Modeling and evaluating trust network inference. In *Seventh International Workshop on Trust in Agent Societies at AAMAS 2004*, 2004.

[27] Li Ding, Lina Zhou, and Timothy Finin. Trust based knowledge outsourcing for semantic web agents. In *Proceedings of IEEE/WIC International Conference on Web Intelligence*, 2003.

[28] Marek J. Druzdzel. Verbal uncertainty expressions: Literature review. Technical report, CMU, 1989.

[29] Patrick Hayes (Eds.). Rdf semantics (w3c recommendation, 10 february 2004). http://www.w3.org/TR/2004/REC-rdf-mt-20040210/, 2004.

[30] Jenny Edwards, Kevin S. McCurley, and John A. Tomlin. An adaptive model for optimizing performance of an incremental web crawler. In *World Wide Web*, pages 106–113, 2001.

[31] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder. Summary cache: a scalable wide-area Web cache sharing protocol. *Proceedings of ACM SIGCOMM'98*, pages 254–265, 1998.

[32] Tim Finin and Anupam Joshi. Agents, trust, and information access on the semantic web. *ACM SIGMOD Record*, 31(4):30–35, 2002.

[33] M.S. Fox and J. Huang. Knowledge provenance: An approach to modeling and maintaining the evolution and validity of knowledge. Technical report, University of Toronto, 2003.

[34] Diego Gambetta, editor. *Trust: Making and Breaking Cooperative Relations*. Department of Sociology, University of Oxford, 2000.

[35] Ed Gerck. Toward real-world models of trust: Reliance on received information, 1998.

[36] Yolanda Gil and Varun Ratnakar. Trusting information sources one citizen at a time. In *Proceedings of International Semantic Web Conference 2002*, pages 162–176, 2002.

[37] Jennifer Golbeck, Bijan Parsia, and James Hendler. Trust networks on the semantic web. In *Proceedings of Cooperative Intelligent Agents*, 2003.

[38] Tyrone Grandison and Morris Sloman. A survey of trust in internet application. *IEEE Communications Surveys Tutorials (Fourth Quarter)*, 3(4), 2000.

[39] Tyrone W. A. Grandison. *Trust Management for Internet Applications*. PhD thesis, Imperial College, University of London, 2003.

[40] R. Guha. Open rating systems, 2004.

[41] R. Guttman, A. Moukas, and P. Maes. Agent-mediated electronic commerce: A survey. *Knowledge Engineering Review*, 13(2):147–159, 1998.

[42] Siegfried Handschuh and Steffen Staab. Cream: Creating metadata for the semantic web. *Comput. Networks*, 42(5):579–598, 2003.

[43] A hanneman. Introduction to social network methods. `http://faculty.ucr.edu/~hanneman/SOC157/TEXT/TextIndex.html`.

[44] Taher Haveliwala. Efficient computation of pageRank. Technical Report 1999-31, Stanford University, 1999.

[45] Jerry R. Hobbs, George Ferguson, James Allen, Richard Fikes, Pat Hayes, Drew McDermott, Ian Niles, Adam Pease, Austin Tate, Mabry Tyson, and Richard Waldinger. A daml ontology of time. `http://www.cs.rochester.edu/~ferguson/daml/daml-time-nov2002.txt`, 2002.

[46] Daml ontology library. `http://www.daml.org/ontologies/`.

[47] Ontaria, by w3c. `http://www.w3.org/2004/ontaria/`.

[48] Semwebcentral, by infoether and bbn. `http://www.semwebcentral.org/`.

[49] Schema web. `http://www.schemaweb.info/`.

[50] Semantic web search, by intellidimension. `http://www.semanticwebsearch.com/`.

[51] Junling Hu and Michael P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 242–250. Morgan Kaufmann Publishers Inc., 1998.

[52] A. Hunter, editor. *Uncertainty in Information Systems*. McGraw-Hill, 1996.

[53] Anthony Hunter and Simon Parsons, editors. *Applications of Uncertainty Formalisms*. Springer, 1998.

[54] Eero Hyvonen. The semantic web – the new internet of meanings. In *Semantic Web Kick-Off in Finland: Vision, Technologies,Research, and Applications*, 2002.

[55] H.Zhuge and P. Zheng. Ranking semantic-linked network. In *www 2003*, 2003.

[56] N. R. Jennings, P. Faratin, M. J. Johnson, P. O'Brien, and M. E. Wiegand. Agent-based business process management. *International Journal of Cooperative Information Systems*, 5(2,3):105– 130, 1996.

[57] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[58] Cliff Joslyn and Luis Rocha. Towards a formal taxonomy of hybrid uncertainty representations. *Information Sciences*, 110(3-4):255–277, 1998.

[59] Audun Jsang. Prospectives for modelling trust in information security. In *Proceedings of Australasian Conference on Information Security and Privacy*, 1997.

[60] L.P. Kaelbling, M.L. Littmna, and A.W. Moore. Reinforcement learning a survey. *Journal of AI Research*, 4:247–285, 1996.

[61] Lalana Kagal. Rei: A policy language for the me-centric project. Technical Report HPL-2002-270, HP Labs, 2002.

[62] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the Twelfth International World Wide Web Conference*, 2003.

[63] Beverly K. Kanh, Diane M. Strong, and Richard Y. Wang. Information quality benchmarks: Product and service performance. *Communications of the ACM*, 45(4):184–192, 2002.

[64] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[65] George J. Klir and Bo Yuan, editors. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995.

[66] Graham Klyne and Jeremy J. Carroll (Eds.). Resource description framework (rdf): Concepts and abstract syntax (w3c recommendation, 10 february 2004). http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, 2004.

[67] Marja-Riitta Koivunen and Eric Miller. W3c semantic web activity. In *Semantic Web Kick-Off in Finland: Vision, Technologies,Research, and Applications*, 2002.

[68] P. Krause and D. Clark, editors. *Representing Uncertain Knowledge: an artificial intelligence approach*. Intellect Press, 1993.

[69] Ninghui Li, Benjamin N. Grosof, and Joan Feigenbaum. Delegation logic: A logic-based approach to distributed authorization. *ACM Trans. Inf. Syst. Secur.*, 6(1):128–171, 2003.

[70] Maxim Lifantsev. Rank computation methods for Web documents. Technical Report TR-76, ECSL, Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY, November 1999.

[71] Sean Luke, Lee Spector, David Rager, and James Hendler. Ontology-based web agents. In *Proceedings of the First International Conference on Autonomous Agents (Agents97)*, pages 59–66, 1997.

[72] Stephen P. Marsh. *Formalising trust as a computational Concept*. PhD thesis, University of Stirling, 1994.

[73] Philippe Martin and Peter Eklund. Embedding knowledge in web documents. In *Proceedings of the 8th International World Wide Web Conference (WWW8)*, pages 324–341, 1999.

[74] James Mayfield and Timothy Finin. Information retrieval on the semantic web: Integrating inference and retrieval. In *Proceedings of the SIGIR 2003 Semantic Web Workshop*, 2003.

[75] Drew McDermott. Why rdf's reification doesn't work. http://lists.w3.org/Archives/Public/www-rdf-logic/2001Apr/0066, 2001.

[76] D. Harrison McKnight and Norman L. Chervany. The meanings of trust. MISRC Working Paper Series, 1996.

[77] Amihai Motro and Philippe Smets, editors. *Uncertainty Management in Information Systems: From Needs to Solution*. Kluwer, 1996.

[78] Lik Mui. *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*. PhD thesis, MIT, 2002.

[79] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, pages 167–256, 2003.

[80] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[81] Simon Parsons. Current approaches to handling imperfect information in data and knowledge bases. *Knowledge and Data Engineering*, 8(3), 1996.

[82] Simon Parsons and Anthony Hunter. A review of uncertainty handling formalisms. In *Applications of Uncertainty Formalisms*, 1998.

[83] Y. Peng and J. Reggia. *Abductive Inference Models for Diagnostic Problem Solving*. Springer-Verlag, 1990.

[84] Dean Povey. Trust management, 1999.

[85] Stphane Lo Presti, Mark Cusack, and Chris Booth. Trust issues in pervasive environments(deliverable wp2-01), September 2003.

[86] Lars Rasmusson and Sverker Jansson. Simulated social control for secure internet commerce. In *Proceedings of the 1996 New Security Paradigms Workshop*, 1996.

[87] R.T. Reagan, F. Mosteller, and C.Youtz. Quantitative meanings of verbal probability expressions. *Journal of Applied Psychology*, 74(3):433–442, 1989.

[88] R.Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.

[89] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *Proceedings of the Second International Semantic Web Conference*, 2003.

[90] G. Salton and M. J. McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[91] Nigel Shadbolt. A matter of trust. *IEEE Intelligent Systems*, 17(1):2–3, 2002.

[92] Chris Sherman. Meta search engines: An introduction. `http://searchenginewatch.com/searchday/article.php/2160771`, September 2002.

[93] Chris Sherman. Metacrawlers and metasearch engines. `http://searchenginewatch.com/links/article.php/2156241`, March 2004.

[94] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993.

[95] Philippe Smets. Varieties of ignorance and the need for well-founded theories. *Information Sciences*, 57-58:135–144, 1991.

[96] Philippe Smets. Probability, possibility, belief: Which and where. *Quantified Representation of Uncertainty and Imprecision*, 1:1–24, 1998.

[97] Michael J. Smithson, editor. *Ignorance and Uncertainty: Emerging Paradigms*. Springer Verlag, 1989.

[98] John F. Sowa. Conceptual graphs summary. *Conceptual structures: current research and practice*, pages 3–51, 1992.

[99] M.B. Twidale and P.F. Marty. An investigation of data quality and collaboration. Technical Report ISRN UIUCLIS–1999/9+CSCW, UIUC, 1999.

[100] Wiebe van der Hoek and Michael Wooldridge. Towards a logic of rational agency. *Logic Journal of the IGPL*, 11(2):133–157, 2003.

[101] Yair Wand and Richard Y. Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):86–95, 1996.

[102] Richard Wang, Veda Storey, and Christopher Firth. A framework for analysis of data quality research. *IEEE Transactions on Knowledge and Data Engineering*, 7(4):623–639, 1995.

[103] Richard Wang and Diane Strong. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information System*, 12(4):5–33, 1996.

[104] L. Xiong and L. Liu. Building trust in decentralized peer-to-peer electronic communities. In *Proceedings of the Fifth International Conference on Electronic Commerce Research*, 2002.

[105] R. Yahalom, B. Klein, and T. Beth. Trust relationships in secure systems– a distributed authentication perspective. In *Proceedings of the 1993 IEEE Symposium on Research in Security and Privacy*, 1993.

[106] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. Collaborative reputation mechanisms in electronic marketplaces. In *HICSS*, 1999.

[107] P. Zimmermann. *PGP User's Guide*. MIT Press, 1994.

[108] Youyong Zou, Tim Finin, Li Ding, Harry Chen, and Rong Pan. Using semantic web technology in multi-agent systems: a case study in the taga trading agent environment. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pages 95–101. ACM Press, 2003.