A Service Search Engine for the Industrial Digital Ecosystems

Hai Dong, *Student Member, IEEE,* Farookh K. Hussain, and Elizabeth Chang, *Senior Member, IEEE*

***Abstract*—Digital Ecosystem (DE) is comprised of heterogeneous and distributed species which can play the dual role of service provider and service requester. Today DE lacks semantic search support, which means that it cannot provide a reliable and trustworthy link between service providers and service requesters. To solve this issue, we design a conceptual framework of a service-ontology-based semantic service search engine. Apart from the function of service search with a novel search model, this framework also provides a quality-of-services (QoS)-based service evaluation and ranking methodology. To evaluate the feasibility of our framework, we implement a prototype in the transport service domain, and compare the performance of the search model with three traditional information retrieval models. The conclusion to this evaluation and suggestions for future works are provided in the final section.**

***Index Terms*—Digital Ecosystems, QoS ranking, semantic service search, service evaluation**

I. Introduction

DIGITAL Ecosystem (DE) is defined as "*an open, loosely coupled, domain-clustered, demand-driven, self-organizing and agent-based environment, in which each species is proactive and responsive for its own benefit and profit*" [6], which is a neoteric terminology whose emergence is a result of the natural existence of the business ecosystem, along with the evolution of business network and information technology. The goal of DE is to improve the efficiency of the communication between internal agents, and to structuralize the existing business ecosystem [18]. The contemporary DE researches focus on theoretical study and application development [9]. DE is composed of two basic elements: species and environments [6]. Species are mainly categorized into three types: biological species, economic species and digital species. Species can play two different roles in DE, as a service requester (client) that needs services, and as a service provider (server) that provides services. Moreover, a given species can be both a service requester and a service provider at the same time, and thus interacts with other species to form a dense social network [6].

DE comprises heterogeneous and distributed species [6].These species are distributed over the existing network infrastructures. Currently available services are less semantic in the network. There is no QoS information available in the network for these services. Thus, it is not possible for a given service requester to know all the other service providers based on a service request. That is why we plan to design a semantic search engine to build a reliable and trustworthy link between service providers and service requesters in the DE environment.

Next, by means of the following case study, we will analyze the service search issues in the field of DE.

As an example, let us assume that John lives in City A and desires a horse transportation service provided by a local competitive company, in order to help him to move horses from City A to City B. From the perspective of the internet services, there are two primary categories of service search engines that can be utilized by John.

The first category is that of generic search engines, such as Google^{TM} and Yahoo! search engine. John can enter "*horse moving companies in A*" into a generic search engine such as Yahoo!. From the retrieved results from the search engine, it is observed that most of the retrieved results do not match John's search intention – horse transportation companies in City A, and the service information is difficult to distinguish and identify from the results. Thus, it could be deduced that the performance of generic search engines is poor in this service search case study.

We can conclude that the performance of generic search engines is due to the following:

1) Generic search engines use traditional keywords-based search strategies without incorporating or taking into account semantic web technologies to assist these search engines to fully understand the sense of user's query words. This causes the lack of precision and hence poor performance of these search engines.
2) Generic search engines are not especially designed for the purpose of service searching. As a result of this, the search process has to be carried out against a much larger information source. Due to this reason and due to the fact that the search process is keyword-based, the retrieved search results are not accurate and do not take into consideration the context of service queries.

Manuscript received January 24, 2009, accepted July 28, 2009.

The authors are with the Digital Ecosystems and Business Intelligence Institute in Curtin University of Technology, Bentley, WA 6102, Australia (phone: 61-0(8)-9266-9270; fax: 61-0(8)-9266-7548; e-mail: {hai.dong, farookh.hussain, elizabeth.chang}@ cbs.curtin.edu.au).

3) The format of the retrieved service information is not standardized, which makes it difficult for users to read and comprehend the retrieved service information.

An enhanced approach is that John can gain access to a repository of local business directory, such as Yahoo!'s or Google^{TM}'s local search, online Yellowpages$^{®}$. These local search engines (here the example is Australian Online Yellowpages$^{®}$) normally can provide John with two service search options as follows:

1) One option is that John can browse businesses under the "*horse*" category in the location "*City A*", by following the "*browse by category*". This approach can provide John with more precise search results and structured service information. The disadvantage is that John needs to follow the whole category of the website step by step, which is expensive in terms of time and effort.
2) Another option is that John can directly enter "*horse moving*" into the business type box and "*City A*" into the location box of the search engine provided by the website. This can save its searching time, but this approach has its own disadvantages as well – the search engine cannot understand the user's query intention and thus returns non-relevant results. Similar to the generic search engines, the reason for this is that local search engines do not use semantic web technologies to help users to denote their searching concepts.

Apart from the lack of semantic web technologies' support, another limitation of the local service search engines is that John cannot find out which company performs best in relation to the provision of horse transportation services. The reason is that these search engines do not provide user-oriented evaluation and ranking mechanisms based on the QoS provided by these services.

Based on the above case study, it is observed that both of the generic and local search engines are far from perfect, when searching for a given service. To solve the issues in this field, we propose to design a semantic service search engine integrated with an innovative QoS technology for the DE environment.

The rest of this paper is organized as follows: first, we will conduct a survey in the field of semantic search, as well as service search and ranking, to analyze the research issues in each field, by comparing the researches with the urgent requirement of semantic service search integrated with QoS ranking and evaluation in the DE environment; next, against these issues, we will present a conceptual model of a semantic service search engine, and discuss its technical details; subsequently, we will reveal an implemented prototype and use the horse transportation example to explain the function of the system; later, we will execute a series of experiments to evaluate the system framework; finally, based on the evaluation results, we will draw conclusions and plan future works.

II. RELATED WORKS

In this section, we will briefly review the researches with regard to semantic search engine, service search and ranking, and analyze the issues within them.

A. Semantic Search Engines

It is well-known that search engines weight web objects with user queries and return to users a ranked list with desired objects. However, in normal cases, the user-desired objects may not be priority-ranked in the list [4] [5] [10] [11] [13] [21]. One reason is that the search engine cannot understand the meaning of user queries. Technologies from the semantic webs can be adopted to enrich the semantic extent of the interactions between search engines and users [26]. Currently, couples of semantic search engines are designed and implemented in order to adapt to different working environments, and the mechanisms that realize these search engines are thus distinct.

Lee and Tsai designed an interactive semantic search engine which collects feedbacks by means of selection in order to better capture users' personal concepts [16]. The search algorithm utilized in this model is an "*iteratively cyclic mechanism*", which includes selecting upper classes and generating lower classes. Selecting upper classes refers to searching for the most suitable web pages by selecting the individuals from the current population and forming a new population; generating lower classes refers to the application of a "*genetic operator*" on upper classes to generate new lower classes.

Chiang et al. presented a semantic search engine based on the smart web query (SWQ) method for web data retrieval [7]. The SWQ architecture contains three main parts: a SWQ search engine and its subcomponents: a "*query parser*" and a " *context ontology determination engine*"; context ontologies for bounding applied domain; a semantic search filter to improve search precision based on retrieving term properties in context ontologies.

Guha et al. delivered a semantic search engine in TAP – a comprehensive semantic web system [15]. The query language for semantic search in TAP is called GetData interface, which allows programs to visit properties of a resource in a semantic graph. Each graph is referenced by a URL, and GetData specifies resource name and property name to access the value of property. Two additional search interfaces are provided by TAP, which are "*search*" that searches for any properties with titles containing a given string, and "*reflection*" that searches for coming and outgoing tracks for a given node in a semantic graph.

Bhagwat and Polyzotis proposed a semantics-based file system search engine – Eureka, which uses an inference model to build the links between files and a FileRank metric to rank the files according to their semantic importance [2]. Eureka consists of two main parts: (1) a crawler which extracts files from file systems and generates two kinds of indices: keywords' indices that record the keywords from crawled files, and a rank index that records the FileRank metrics of the files; (2) when search terms are entered, the query engine will match the search terms with keywords' indices, and determine the matched file sets and their ranking order by an information retrieval-based metrics and FileRank metrics.

Dichev and Dicheva exhibited a view-based semantic search engine in the context of a topic-centered learning repository, by means of the extension of the Topic Maps (TM) model which is a lightweight ontology model consisting of topics and relationships between topics [8]. The environment where TM is implemented is TM4L, which is "*an environment for building and using ontology-aware learning repositories represented by topics*". A view in the TM is defined as a collection of related topics, occurrences of topics, associations between topics and scopes of topics. The view-based semantic search in the TM4L environment includes two phases: transforming a view-based query to a traversal expression and then locating some corresponding resources; using the retrieved resources to locate other relevant resources.

Wang et al. projected a semantic search methodology to retrieve information from normal tables, which has three main steps: identifying semantic relationships between table cells; converting tables into data in the form of a database; and, retrieving objective data by query languages [20]. The research objective defined by the authors is how to use a given table and a given domain knowledge to convert a table into a database table with semantics. The authors' approach is to denote the layout by layout syntax grammar and match these denotations with given templates which can be used to analyze the semantics of table cells. Then semantic preserving transformation is used to transform tables to database format.

Takama and Hattori designed a hybrid search engine which integrates the function of resource description framework (RDF) file search and normal document search. When a document is registered into a repository, its registration information is stored in the form of Dublin Core metadata. Thus, the search engine can use the function of metadata search to retrieve normal documents [24].

Here, in contrast to the requirement that a semantic service search engine is desired, the above search engines show some disadvantages because there is not any search engine specially designed for retrieving service information. Unlike normal file retrieval, service information needs to: 1) be easily understood by users; and 2) fully satisfy users' service requests. The first requirement needs service information to be standardized and clearly defined. The second requirement needs service information to be semanticized, in order to make a comparison with users' requirements. In addition, according to the case study shown in the first section, service information should have the capability of being quantified to satisfy a user's request – the one that is best for me. From this perspective, current semantic search engines cannot provide such service repository, use semantic technologies to understand users' service requests, and support service ranking based on personalized QoS criteria.

B. *Service Search and Ranking*

While there are a great number of semantic search engines being developed (e.g., SWoogle, TAP), few of them attempt to provide optimized solutions for the service search field.

Liu et al. [17] developed an e-service platform integrated with semantic search for e-service metadata. E-service metadata refers to the descriptions of e-services and providers, which is adopted to publish and to discover e-services. There are two types of metadata in the system: business level metadata – the descriptions of e-service providers, and service level metadata – the descriptions of basic information about e-services. The authors adopt Universal Description, Discovery and Integration (UDDI) which is a web service standard to register and search e-services. Three means for searching service and business are provided, which are find_business, find_service and XQuery. Find_business returns a list of service providers for specific conditions; find_service returns the information for a list of services who match customized conditions; XQuery queries extended metadata added to a businessService list.

The limitations of the e-service search engine can be described as follows:

1) A one-tier (service categories-services) only concept hierarchy cannot reflect the complex relationships between services in the DE environment;
2) There is no methodology provided for the concept hierarchy update in order to adapt to the changes in service environment;
3) The volume of its knowledge base seems so limited that it can be applied only to limited fields;
4) There is no ranking methodology provided for the querying results, which could lead to unorganized data structure and poor presentation to the user.

While a large amount of literature focuses on evaluating quality of services (QoS) [23], few of them study the integration of service evaluation and service ranking system for service retrieval in the DE.

Toma et al. [19] propose a web service ranking system based on two different ranking strategies. One strategy is to use the Web Service Modeling Ontology (WSMO) to describe the values of Non-Functional Properties (NFPs) of web services, such as QoS, Service Level Agreement (SLA). Hence, web services can be ranked according to the values of user-preferred NFPs. Another strategy is a multi-criteria ranking, which considers ranking multiple NFPs from three main perspectives – the user-preferred NFPs, the level of importance of the NFPs, and the ascending or descending order of services.

Gekas [14] proposes a set of metrics for web service ranking. Four main categories of ranking strategies are provided by these metrics, which are degree-based rankings that calculate the percentage of fed services in each web service, hubs-authorities-based rankings that calculate the ratio between the number of incoming services and the number of outgoing services, non-functional rankings that focus on the NFPs of web service, and non-connectivity rankings that focus on the connectivity of web service networks.

The limitations of these service ranking systems can be concluded as follows:

1) They are all designed for the web service or SOA (Service Oriented Architecture) [22] environment which cannot adapt to the broader and more complicated DE;
2) Despite all of them being equipped with the QoS ranking methodology, none of them integrates its ranking systems with the corresponding QoS evaluation systems, which cannot ensure the correctness of the QoS data;
3) None of them considers the factor of trustworthiness and reputation in the service ranking;
4) None of them attempts to implement its service ranking methodologies into service search engines.

As can be clearly seen from the discussion above, research is being carried out independently in both the fields of semantic service search and ranking, without any attempt to integrate them together.

III. System Design

In order to address the issues discussed in the last section, this research proposes a semantic service search engine enabling the service requester to retrieve services based on domain-specific QoS criteria in the DE environment, which integrates the notion of semantic search and QoS. In the following sections, we will provide the overall system architecture, and explain each component in detail.

A. *Overall System Architecture*

The overall system architecture of the semantic service search system can be found in Fig. 1. The system consists of four primary components, which are service knowledge base, service reputation database, service search module, and service evaluation module. Service knowledge base is to store domain-specific service ontologies and service description entity (SDE) metadata. Service reputation database is to store domain-specific criteria and reputation values for each service. Service search engine is to retrieve SDE metadata for service requesters. Service evaluation module is to evaluate a service provider's reputation after a service transaction.

The whole workflow of the system is as follows:

1) First of all, a service requester enters a set of key terms into the search engine interface;
2) Then the service search module will retrieve service metadata from the service knowledge base;
3) Following that the retrieved service metadata will be ranked based on multi-linear ways – service providers' reputation values, or domain-specific evaluation criterion values which are both stored in the service reputation database;
4) After the service requester completes a service transaction with a service provider, the service provider will then be required to send an email to the system administrator to confirm that the service requester obtains the right to evaluate the quality of the service transaction;
5) Once the system administrator has received this confirmation email, the service requester will be permitted to log in to the service evaluation interface to perform the evaluation.

In the following sections, we will illustrate in detail the architecture and function of each component.

B. *Service Knowledge Base*

As described previously, the purpose of a service

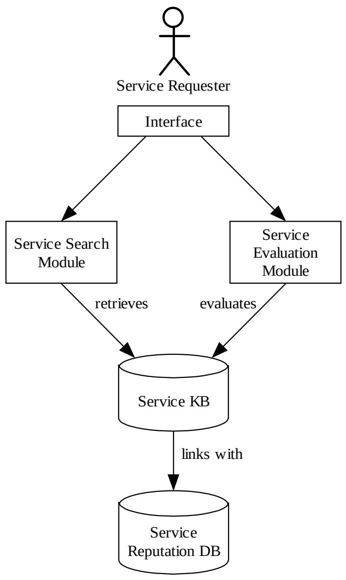


Fig. 1. Overall system architecture of semantic service search engine.

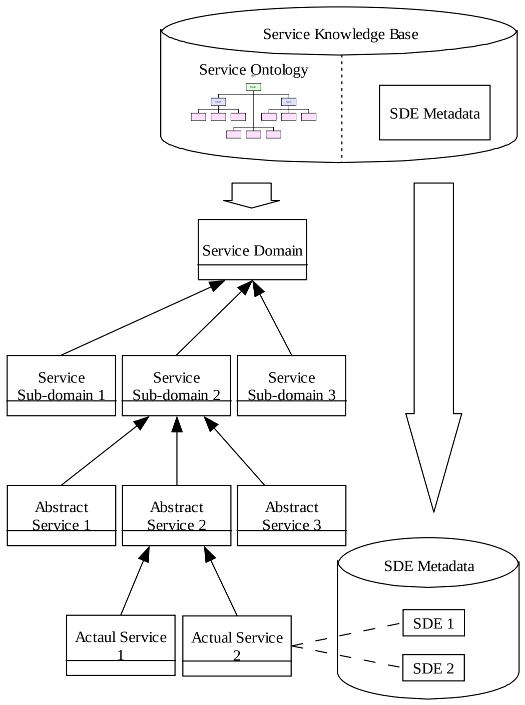


Fig. 2. System architecture of service knowledge base.

knowledge base is to store service ontology and SDE metadata, in which the semantically related (actual) ontological concepts and SDE metadata are linked by referencing their URIs to each other. It is noted that there are two rules contained in the semantic relationship as follows:

1) A service concept may semantically relate to arbitrary SDE metadata;
2) A SDE metadata may also semantically relate to arbitrary service concepts.

The system architecture of the service knowledge base is shown in Fig. 2.

The structure of the service ontology concepts is a four-layer hierarchy. The first layer is the root of the hierarchy, which represents the abstract concept of all services in a domain. The second layer is the preliminary specialization of the abstract service concept, which represents the service sub-domain concepts. The third layer is the further specialization of the abstract service concepts, which represents the abstract service concept in each sub-domain. The bottom layer is the concrete service concept, which corresponds to the actual services in the real social environment. The main difference between the abstract concepts and concrete concepts is that only the latter can link to SDE metadata.

The Service Ontology is defined as the conceptualization of the Service, which is identified by a Service Name, defined by a Service Description.

We present the Service Ontology as the combination of the ontology name and a tuple where the elements of the tuple can be complex elements as defined below:

Service [Service Name, Service Description] where

Service Name refers to the name that can be used to uniquely identify a service.

Service Description refers to the definitional descriptions of a service. The normal form of a service description is a set of words (noun, adjective, or adverb). A service concept may have many service descriptions. The purpose of setting the property of service description is to compute the semantic similarity values between service concepts and queries, which will be introduced later.

The service ontology is the definition of the service concept in the root of the service concept hierarchy. As children concepts, all other concepts in this hierarchy automatically inherit its properties. In addition, actual concepts have one extra property defined below:

Linked Metadata refers to the URIs of semantically related SDE metadata to a concept.

The major purpose of SDE metadata is to elicit meaningful information with regard to a service provided by a service provider in the real environment. It is noted that a SDE metadata can be provided only by a service provider, which means that even the same service provided by two service providers are conceptualized to two SDE metadata.

The SDE metadata can be represented as a tuple where the elements of the tuple can be complex elements as defined below:

[Linked Concepts, Service Provider Name, Provider Address, Provider Contact Details, SDE Description] where

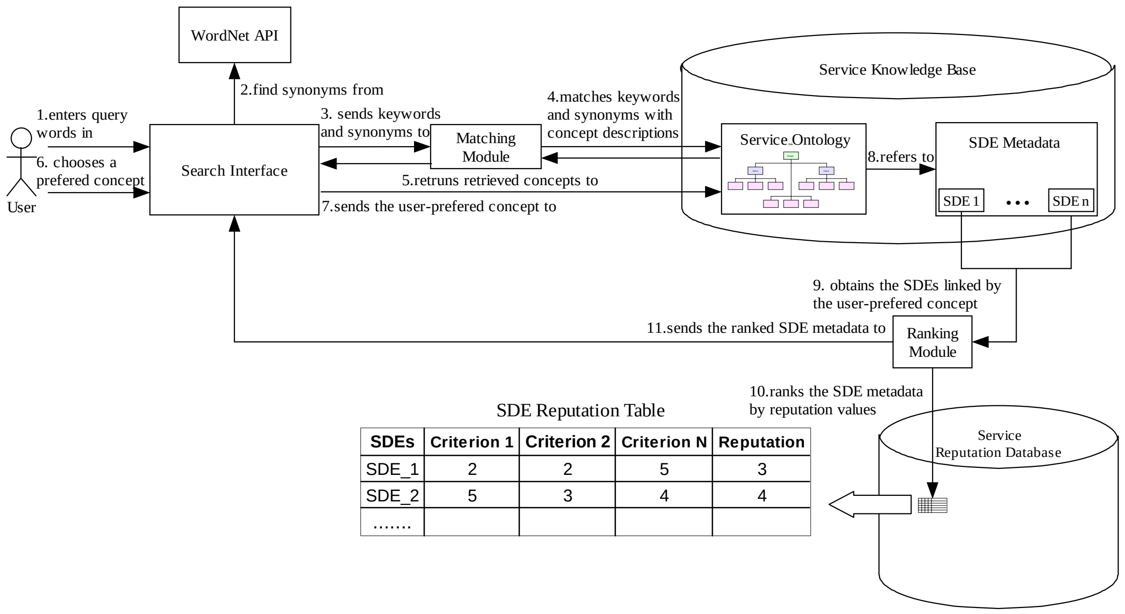
Linked Concepts refers to the URIs of semantically related concepts to the SDE metadata.

Service Provider Name refers to the name of the person or organization that provides a service.

Provider Address refers to the address where a service provider can be located.

Provider Contact Details refer to the information regarding how a service provider is contacted, for instance, mail box, phone number, fax number, website and so on.

SDE Description refers to the detailed text description with regard to the content of a service. This can be used for matching with a service concept, which will be described later.


WordNet API
2.find synonyms from
1.enters query words in
6. chooses a prefered concept
User
Search Interface
3. sends keywords and synonyms to
Matching Module
4.matches keywords and synonyms with concept descriptions
5.retruns retrieved concepts to
7.sends the user-prefered concept to
Service Knowledge Base
Service Ontology
8.refers to
SDE Metadata
SDE 1
SDE n
9. obtains the SDEs linked by the user-prefered concept
11.sends the ranked SDE metadata to
Ranking Module
10.ranks the SDE metadata by reputation values
Service Reputation Database
SDE Reputation Table

SDEs	Criterion 1	Criterion 2	Criterion N	Reputation
SDE_1	2	2	5	3
SDE_2	5	3	4	4
…….				

Fig. 3. System architecture of semantic service search module.

C. *Service Search Module*

The system architecture of the service search module is shown in Fig. 3. The workflow of the search engine is as follows:

First of all, a user enters a set of query terms mixed with Boolean operators (e.g. and, or, not) into the search interface. The search interface will send each query term to the WordNet API. If one query term can be retrieved from the API, the API will return its synonyms; otherwise, the query term will be filtered. After the process has been completed, the search interface will send the query terms and their synonyms to the matching module. A matching algorithm is run by the matching module to compute the similarity values between the service ontology concepts stored in the service knowledge base and the query terms. The concepts with higher similarity values will be returned to the search interface and ranked according to their similarity values. The user then can choose the concept which is closest to his/her query intention. Once the user decides a concept, all its semantically relevant SDE metadata will be retrieved from the service knowledge base and sent to a ranking module. The ranking module will obtain the QoS information regarding the retrieved SDE metadata from the service reputation database. Then both of the SDE metadata and their QoS information will be sent back to the search interface. The user may rank the SDE metadata based on various QoS options.

ECBR Model

To realize the function of computing similarity values between actual ontological concepts and queries, we design an ECBR (Extended Case-based Reasoning) algorithm. The principle of the ECBR algorithm is to compare a group of query terms and their synonyms with the service descriptions of a service concept. For a service description, if a query term is contained in it, a value 1 will be awarded; if a synonym of a query term is contained in it, a value 0.5 will be awarded. After the comparison process, the sum of the values for the comparison of whole query terms will be normalized by the length of the terms in the service description; thus, its value should be between 0 and 1. Since one concept may have more than one service descriptions, the maximum value among them is the similarity value between the concept and the group of query terms. The ECBR algorithm for computing the similarity between a concept C and a query Q is mathematically shown below:

$$sim(C,Q)=\max_{SD_i\in C}\left(\sum_{t_{ih}\in SD_i}\frac{match(t_{ih},Q)+match_s(t_{ih},S)}{l_{SD_i}}\right) \quad (1)$$

with

$$match(t_{ih},Q)=\begin{cases}1 & if\ \exists q_t\left|(t_{ih}=q_t)\wedge(q_t\in Q)\right.\\ 0 & otherwise\end{cases} \quad (2)$$

$$match_s(t_{ih},S)=\begin{cases}0.5 & if\ \exists s_k\mid(t_{ih}=s_k)\wedge(s_k\in S)\\ 0 & otherwise\end{cases} \quad (3)$$

where SD_i is a service description of the concept C, t_{ih} is a term that occurs within SD_i, and l_{SDi} is the frequency of all terms appearing in SD_i; q_t is a term that occurs within the query Q, and S are synonyms of Q, which consists of a group of terms s_k.

The ECBR model is very simple to implement, and it does not need to generate index terms before matching, which saves the preprocessing time. It can also adapt to the flexibility of concepts that often needs regenerating index terms in most of index term-based algorithms. Since the algorithm is independent of index terms, it does not have the issue of index term independency. The more in-depth information can be referenced from [12].

D. *Service Reputation Database and Service Evaluation Module*

In this section, we will introduce the theoretical foundation of our QoS evaluation, as well as the conceptual model of service reputation database and service evaluation module.

CCCI Metrics

It is believed that semantic web technology can improve the quality of QoS inspection since enhanced semantic descriptions can reason QoS parameters [25]. The basis of service reputation evaluation, in this semantic search engine, is based on the theory of CCCI Metrics. CCCI Metrics is a group of QoS metrics developed by Chang et al. [3], as a means of measuring a service requester’s trust to a service provider after a service transaction. These metrics are grounded on the assumption that a service interaction involves at least one criterion. A service can be regarded as an ordered set of criteria. A criterion is defined as a decisive factor of the mutually agreed service performance between the service provider and service requester. A criterion could be an industry standard adopted by professional bodies or domain experts. Therefore, the service requester can evaluate the performance of the service provider in each of the decisive factors after the service interaction. Subsequently, the issue of determining the Quality of Service (QoS) comes down to the issue of measuring and quantifying the delivered service according to each individual criterion and their aggregation.

We would make use of the proposed CCCI metrics and apply them in the scenario of service retrieval in order to rank service providers. The workings of the proposed application of CCCI metrics in the domain of service retrieval are explained in the rest of this section with the help of a case study. It is important to note that in the context of this paper, it is not possible to explain the workings of the CCCI metrics in detail. We would like to encourage interested readers to refer to Chang et al. [3] for an in-depth explanation of the workings of CCCI metrics.

In the instance of John who wants to retrieve and evaluate a horse moving service in City A, the metrics proposed within the framework of CCCI Metrics’ which would help John to rank the service providers are as follows:

1) To find out the reputation values (0 to 6) of the companies that provide a horse transportation service in City A from the perspective of horse moving service – *reputation*;
2) To find out the performance (0 to 6) of each company against the industry standards (criteria) of horse transportation services – $ABCorr_{Criterion}$;
3) To evaluate the level of John's trust (0 to 6) in a horse moving service – *trustworthiness*;
4) To evaluate its performance (0 to 6) against each horse moving industry standard (criterion) – $ABCorr_{Criterion}$;
5) To evaluate the clarity (0 to 1) of each horse transportation service industry standard (criterion) – $Clear_{Criterion}$;
6) To evaluate the importance (1 to 3) of each horse transportation service industry standard (criterion) – $Imp_{Criterion}$.

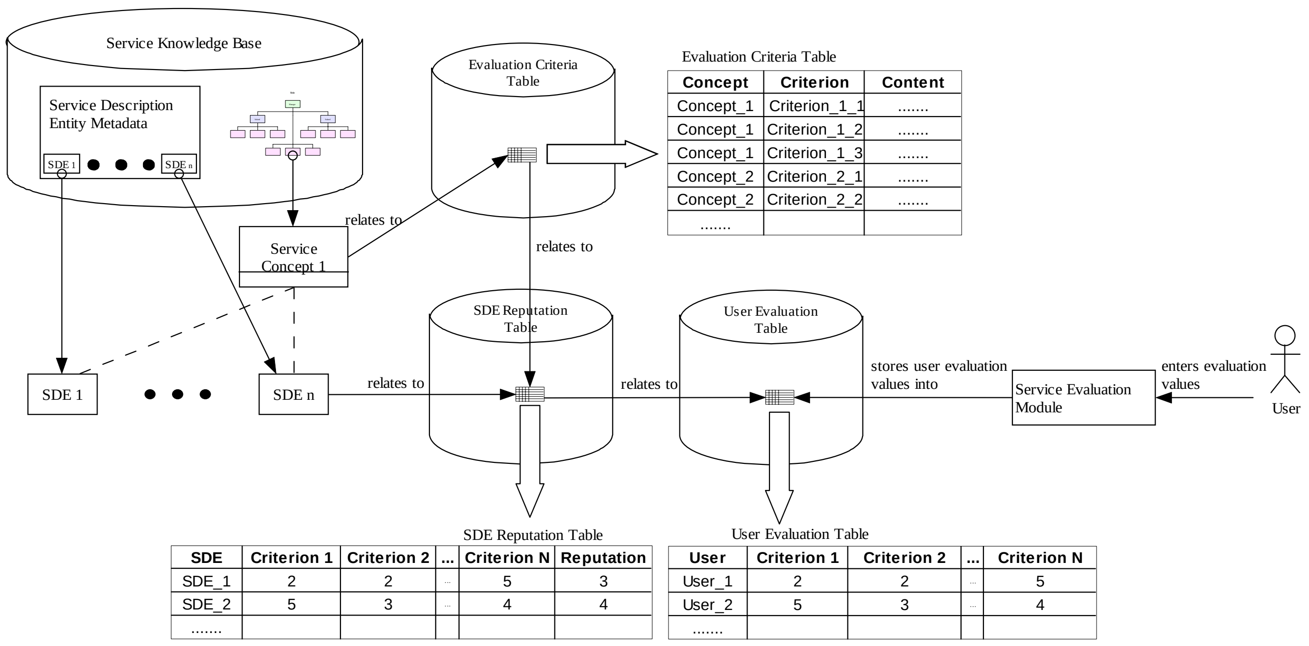
Service Reputation Database

The service reputation database consists of three kinds of tables as follows:

1) Evaluation criteria table. This table stores QoS evaluation criteria information for each actual concept in a service ontology. These criteria are concept-specific. In other words, even though there is the same criterion name and contents for two concepts, the semantics of these criteria are different and thus the metadata under the concepts cannot be compared together. There is only one table in the service reputation database.
2) SDE reputation tables. These tables store the reputation and criterion values of all semantically related SDE metadata for each (actual) service concept. Each table corresponds to one service concept and the counterparts in the evaluation criterion table. Users can use the data in the tables to rank SDE metadata under corresponding service concepts.
3) User evaluation tables. Each table records all users' evaluation values for an SDE metadata in the semantics of a concept, which corresponds to a row in a SDE reputation table that relates to this concept. In other words, the data of a row in a reputation table are computed by a corresponding user evaluation table.

Service Evaluation Module

The service evaluation module is the application of a database management system (DBMS), which is built upon the service reputation database. When a service requester evaluates a service (SDE metadata), according to the rule of CCCI Metrics, s/he needs to evaluate the service from the perspective of each criterion that corresponds to the concept to which this metadata semantically relates, as well as the clarity and the importance of each criterion. The evaluation values then will be stored as a new row in the user evaluation table to which the metadata corresponds. The trustworthiness value of the service provider on this service metadata, and the performance value of each criterion towards this metadata then will be calculated and compared with the all other users' data in this user evaluation table, in order to obtain the new reputation value and each criterion's new performance value. Later, the corresponding row in an SDE reputation table will be updated by these values. The updated values will then be utilized for ranking purposes. The system architecture of the service evaluation module can be seen in Fig. 4.



Evaluation Criteria Table

Concept	Criterion	Content
Concept_1	Criterion_1_1	....
Concept_1	Criterion_1_2	....
Concept_1	Criterion_1_3	....
Concept_2	Criterion_2_1	....
Concept_2	Criterion_2_2	....
....		

SDE Reputation Table

SDE	Criterion 1	Criterion 2	...	Criterion N	Reputation
SDE_1	2	2	...	5	3
SDE_2	5	3	...	4	4
....					

User Evaluation Table

User	Criterion 1	Criterion 2	...	Criterion N
User_1	2	2	...	5
User_2	5	3	...	4
....				

Fig. 4. System architecture of service reputation database and evaluation module.

IV. Prototype Implementation

In this section, we provide the implementation details concerning the semantic service search engine prototype.

A. Service Knowledge Base and Service Reputation Database Implementation

The prototype of service knowledge base is built in Protégé-OWL. As mentioned above, the service knowledge base consists of domain-specific service ontologies and SDE metadata. With regard to the service ontology, we choose transport service as the domain for the ontology design. It is crucial to note that the domain ontology should be designed by domain experts and updated in a community-based manner. To acquire the domain knowledge and to design the transport service ontology, we conducted a general survey of over 1000 Australian transport service companies' websites and referred to several transport service classification systems from Open Directory Project (www.dmoz.org), Kompass (www.kompass.com) etc. The detailed information regarding the transport service ontology can be found in [12]. We also propose a community-based ontology evolution model which allows users to change the ontology on a voting-based manner. The voting result is determined by incorporating the ones from the domain expert group and the normal user group.

With regard to the SDE metadata, we select all business webpages under the transport service category of Australian Yellowpages® (http://www.yellowpages.com.au) website as the resource to collect metadata. A semantic crawler is designed to implement the metadata collecting task, which can refer to [12].

The prototype of the service reputation database is realized by MySQL.

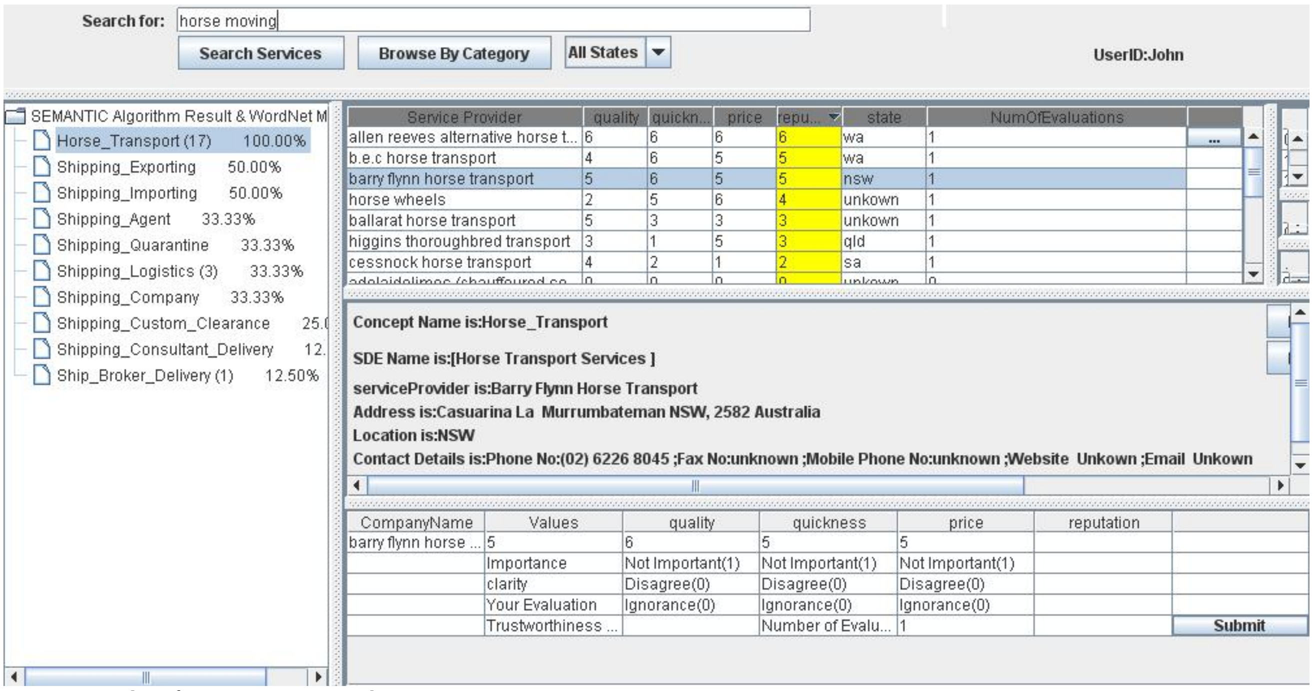
B. Service Search Module and Service Evaluation Module Implementation

In this section, the implementation details with respect to the service search module and service evaluation module will be introduced respectively.

Service Search Module

The prototype of the semantic service search engine is built with JAVA, incorporated with WordNet API, Protégé-OWL API and MySQL API. The screenshot of the semantic service search engine prototype is shown in Fig. 5. The whole search operation process is illustrated as follows.

When a user (John, e.g.) wants to search a horse transportation service, he can directly enter the terms of "*horse moving*" into the search interface, and then click the button "*Search Services*". The matched concepts are displayed in the left side of the search interface, and ranked according to their similarity values with the query terms. If one concept has semantically linked SDE metadata, the amount of SDE metadata will be shown behind the concept name, e.g., "*Horse Transport (17)*" in Fig. 5. Then John can decide which concept is closer to his search intention. In this case, the concept "*Horse Transport*" that ranks in the highest position is obviously the closest one. After John clicks the "*Horse Transport*", the service providers of its associated SDE metadata are displayed on the right side of the search interface. Then, John can rank these services in multiple ways. The default ranking order is the providers' reputation values under the "*Horse Transport*" service concept (highlighted under the label "*reputation*"). Moreover, John can choose the ranking in the alphabetic order of service providers' names or names of states, e.g. "*Service Provider*", "*State*", in the values of each domain-specific evaluation criterion, e.g., "*quality*",


Search for: horse moving
Search Services
Browse By Category
All States
UserID:John
SEMANTIC Algorithm Result & WordNet M
Horse_Transport (17) 100.00%
Shipping_Exporting 50.00%
Shipping_Importing 50.00%
Shipping_Agent 33.33%
Shipping_Quarantine 33.33%
Shipping_Logistics (3) 33.33%
Shipping_Company 33.33%
Shipping_Custom_Clearance 25.(
Shipping_Consultant_Delivery 12.
Ship_Broker_Delivery (1) 12.50%
Submit

Service Provider	quality	quickn...	price	repu...	state	NumOfEvaluations
allen reeves alternative horse t...	6	6	6	6	wa	1
b.e.c horse transport	4	6	5	5	wa	1
barry flynn horse transport	5	6	5	5	nsw	1
horse wheels	2	5	6	4	unkown	1
ballarat horse transport	5	3	3	3	unkown	1
higgins thoroughbred transport	3	1	5	3	qld	1
cessnock horse transport	4	2	1	2	sa	1
adelaidelimos (chauffeured co	0	0	0	0	unkown	0

Concept Name is:Horse_Transport

SDE Name is:[Horse Transport Services]

serviceProvider is:Barry Flynn Horse Transport
Address is:Casuarina La Murrumbateman NSW, 2582 Australia
Location is:NSW
Contact Details is:Phone No:(02) 6226 8045 ;Fax No:unknown ;Mobile Phone No:unknown ;Website Unkown ;Email Unkown

CompanyName	Values	quality	quickness	price	reputation	
barry flynn horse ...	5	6	5	5		
	Importance	Not Important(1)	Not Important(1)	Not Important(1)		
	clarity	Disagree(0)	Disagree(0)	Disagree(0)		
	Your Evaluation	Ignorance(0)	Ignorance(0)	Ignorance(0)		
	Trustworthiness ...		Number of Evalu...	1		Submit

Fig. 5. Screenshot of semantic service search engine prototype.

"*quickness*" or "*price*", or in the number of evaluations for each SDE metadata, e.g., "*NumOfEvaluations*". Once John clicks a service provider ("*Barry Flynn Horse Transport*" in this case), the detailed information regarding the metadata then is displayed in the mid-right part of the search interface.

Service Evaluation Module

The service evaluation interface is bundled with the service search interface. Here we use John's case to illustrate the function and process of the service evaluation prototype.

After John finishes the horse transport service transaction with the service provider – Barry Flynn horse transport, he can send an email to the system administrator to request a service evaluation authentication. Once the system administrator receives John's email, s/he will send an email to the service provider to confirm the completion of John's transaction. When the administrator receives the confirmation email from the service provider, the administrator will insert John's user id as a new row into the service provider's horse transport service user evaluation table. S/he will also send John an email to notify him that he can now evaluate. When John logs into the system and finds the metadata again, he will find there is a service evaluation module that is under the metadata information (Fig. 5). Then he can assign values to the importance, clarity, and evaluation for the three criteria of the horse transport service. Once he decides the values and click the "*Submit*" button underneath, John's trustworthiness value to the service provider will be calculated according to the algorithm contained in the CCCI Metrics. Furthermore, the reputation value and performance value of each criterion of the service provider for this service will be recomputed.

V. Experiment and Results

In this section, we experimentally evaluate the conceptual model of the semantic service search engine. The goals of this experiment are: (1) to evaluate the feasibility of the semantic search engine prototype; (2) to find the proper threshold for the ECBR model; and (3) to compare the performance of the ECBR model with some traditional information retrieval models.

A. *Performance Indicators*

To evaluate our semantic service search engine, five performance indicators from the field of information retrieval are employed, which are Precision, Recall, Mean Average Precision, Harmonic Mean, and Fallout Rate.

Precision in the information retrieval is used to measure the preciseness of a retrieval system [1]. In this experiment, Precision is the proportion of retrieved relevant concepts in all retrieved concepts for a query, which can be represented below:

$$\text{Precision} = \frac{\text{number of retrieved relevant concepts}}{\text{number of retrieved concepts}} \tag{4}$$

Before we introduce the definition of Mean Average Precision, the concept of Average Precision should be defined. Average Precision is the average of precision values after truncating a ranked retrieved concepts list after each of relevant concepts for a given query. This indicator emphasizes the return of more relevant concepts earlier [1], which can be represented as:

$$\text{Average Precision(S)} = \frac{\text{sum(precision @ each relevant concept in the list)}}{\text{number of retrieved relevant concepts in the list}} \tag{5}$$

Mean Average Precision refers to the average of the average precision values for a set of queries, which can be represented as:

$$\text{Mean Average Precision} = \frac{\sum_{i=1}^{n} \text{Average Precision}(S_i)}{n} \tag{6}$$

Recall in the information retrieval refers to the measure of effectiveness of a query system [1]. In this experiment, Recall is the proportion of retrieved relevant concepts in all relevant concepts for a query, which can be represented below:

$$\text{Recall} = \frac{\text{number of retrieved relevant concepts}}{\text{number of relevant concepts}} \tag{7}$$

It is important to note that the number of relevant concepts can be determined only by a peer-reviewed method, as the estimation of relevance between concepts and queries requires detailed knowledge of all concepts in the collection, which can only be manually implemented in the current situation [1].

Harmonic Mean (F-measure) in the information retrieval is used as an aggregated performance scale for the search engine [1]. In this experiment, Harmonic Mean is the mean of Precision and Recall, which can be represented below as:

$$\text{Harmonic Mean} = \frac{1}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \tag{8}$$

When the Harmonic Mean value reaches the highest, it means the integrated value between Precision and Recall reaches to the highest at the same time.

All of the above indicators have the same limitation – they do not consider the number of non-relevant concept in a retrieved collection. In addition, if there is no relevant concept in the retrieved collection, recall cannot be defined. To solve this issue, we need another performance indicator – Fallout Rate. The Fallout Rate for a query is the proportion of retrieved non-relevant concept in the whole collection of non-relevant concepts in an ontology, which is represented as:

$$\text{Fallout Rate} = \frac{\text{Number of retrieved non-relevant concepts}}{\text{Number of non-relevant concepts}} \quad (9)$$

In contrast to other performance indicators, the lower the fallout value is, the better the search engine's performance is.

The following experiment will be executed based on the five performance indicators.

B. Experiment and Results

To evaluate the performance of our ECBR model from the perspective of information retrieval, and to provide alternative ways of concept retrieval, we design three alternative concept retrieval programs based on three index term-based models from the fields of information retrieval – vector space model (VSM), latent semantic indexing (LSI) model and probabilistic model, which are the most typical models respectively from the field of algebra and probability. The mechanisms and algorithms concerning these three models can be referred from [1]. In addition, to obtain the most precise statistical data, we instantiate 100 queries which almost cover every domain under transport service field. All the indicators' results will be averaged by 100.

As described before, three main goals are involved in the experiment, which can be achieved by two tasks:

1) We will use the peer-reviewed method to test the performance of our semantic service search engine on the five indicators, by the level of different threshold values that ranges from 0 to 0.8 with the increment of 0.05. By means of the Harmonic Mean that leverages the two most primary indicators – Precision and Recall, the optimal threshold value for the ECBR model can be determined. This task can realize the first and second goal.
2) It is important to note that the four information retrieval models cannot be compared, unless an optimal threshold value for each model is selected, and then the performance of these models can be viewed and contrasted by the performance indicators at the optimal threshold values. Therefore, we will use the same evaluation method employed in the ECBR model to test the performance of the VSM, LSI and probabilistic model, and also to determine the optimal threshold values for these models based on the Harmonic Mean. Finally, we will horizontally compare the performance of the four models on the five indicators and on their selected optimal threshold values. This task can realize our third goal.

Task 1: Testing the ECBR Model

The testing result of the ECBR model is shown in Table I (See Appendix A). It is observed that along with the increase of the threshold value, the Precision increases sharply from 12.38% to 79.43%). In particular, when the threshold is over 0.55, the Precision jumps from 27.95% to 66.46 %, which is a nearly 40% rise. Mean Average Precision is to test the quickness and precision of a search. Here the Mean Average Precision keeps in a higher level that ranges from 70.97% to 91.15%. In contrast, Recall ranges from 75.26% to 36.62%. Harmonic Mean experiences a curvilinear change, in which the peak is 51.04% at the threshold value 0.55 and 0.6. Fallout is the champion of variation, which reduces to nearly 1/800.

Since the highest Harmonic Mean value turns out on the threshold value of 0.55 and 0.6, and the other parameters are also same on the two threshold values, we choose both of the statistical data as the participants for the forthcoming comparison with the VSM, LSI and probabilistic model.

Task 2: Comparing the Performance of ECBR Model with the VSM, LSI and Probabilistic Model

The testing results of the VSM, LSI and probabilistic model are revealed in Table II to Table IV (See Appendix B to D) respectively. According to their highest Harmonic Mean values, we choose statistical data at threshold value 0.4 as the comparison participant for the VSM and 0.55 for the LSI as well as 0.45 for the probabilistic model. The comparison results are shown in Table V (See Appendix E).

Subsequently, we will horizontally compare the performance of the four models with their participants, based on the five performance indicators:

1) Precision. ECBR>VSM>PM>LSI. The ECBR has no doubt an outstanding performance. Its value is 66.46%, compared with the performance of other models which slightly varies from 40% to 50%.
2) Mean Average Precision. ECBR>LSI>VSM>PM. The champion is the ECBR again, with over 10% gap to the second.
3) Recall. PM>VSM>LSI>ECBR. The differences among these models are small, varying from 41.43% to 44.44%
4) Harmonic Mean. ECBR>PM>VSM>LSI. The ECBR has an overwhelming advantage over the other models on the scale, and is the only model whose value is beyond 50%.
5) Fallout Rate. VSM>PM>LSI>ECBR. As introduced above, the lower the value on Fallout Rate, the better the model. Hence, The ECBR performs best on this factor, which is nearly half of that of the other models.

Discussion

By means of this experiment, it can be deduced that of the four models, the ECBR performs outstandingly, leading in four of the five indicators. Precision and Mean Average Precision are the measurements of search accuracy and quickness. Here, the ECBR shows an overwhelming advantage compared with the other models. Another advantage of the ECBR is shown on the indicator of Fallout Rate, which reveals its low error rate. Compared with the other indicators, the performance of the ECBR is worse on the Recall. It is noted that the other three models also perform poorly on this indicator. One reason for this is that the ECBR's relatively higher threshold (highest among the four models) restricts its performance; in contrast, the ECBR's Recall performs well in the lower threshold values. Another reason is that we choose the threshold based on the Harmonic Mean, which is a mathematical mean between Precision and Recall. In other words, the highest Harmonic Mean value chooses the most balanced Precision and Recall, and thus sacrifices the performance of recall. However, the highest Harmonic Mean value still proves that the ECBR has the best performance in comparison with the three typical information retrieval models.

VI. Conclusion and Future Works

In this paper, we aim to providing a trustworthy and reliable technology for linking service providers and service requesters in the DE environment. DE is comprised of heterogeneous and distributed species, and urgently needs semantic enhancement. However, based on our case study, currently neither generic search engines nor local search engines can satisfy the advanced requirements of DE. We also survey the emerging works in the field of semantic search engines, service search and service ranking, and finally conclude that all of the surveyed designs are far from perfect, and there is no comprehensive system that integrates semantic service search and service evaluation as well as service ranking. Thus, based on the research issues, we design the conceptual framework of a semantic service search engine. The proposed function of this search engine is to enable service requesters to retrieve and evaluate services published by service providers. The four main components of this system include: a service knowledge base, a service reputation database, a service search module and a service evaluation module. The service knowledge base consists of a service ontology and SDE metadata, in which SDE metadata are associated and clustered with the ontology. In the service search module, we design an ECBR model, in order to retrieve ontological concepts for users' queries. The theory of our QoS evaluation is based on Chang et al.'s work – CCCI Metrics [3], as a methodology to measure the trustworthiness of service requesters to services and thus to quantify the reputation degree of service providers. Based on the CCCI Metrics theory, the service reputation database is built by linking three kinds of tables, which are designed for realizing the function of service evaluation and ranking. The service evaluation module utilizes DBMS for service requesters to evaluate services after service transactions. Following that, we implement the prototype of our semantic service search engine in the transport service domain. We create a transport service ontology, and obtain the metadata from the Australian Yellowpages® website. By means of user requirements used in a case study, we explain the functions of this prototype. To evaluate our conceptual framework, we adopt five performance indicators from the field of traditional information retrieval. We compare the performance of the ECBR with the VSM, LSI and probabilistic model. The method is to obtain the optimal threshold value of each model when the harmonic mean value reaches to the highest. Then we horizontally contrast the performance of each model on the five indicators and on the optimal thresholds. The comparison shows that the ECBR model reveals overwhelming advantages in this experiment, compared with other three models. The only disadvantage is that all of the four models perform poorly for the recall indicator. The reason is that the optimal threshold value for each model is determined by the harmonic mean that strives for the most balanced score between precision and recall.

The future works are planned as follows:

1) We are going to use more performance indicators which can better model user requirements.
2) To address the defect of low recall rate that appeared in the experiment, we will modify our ECBR algorithm to obtain better performance.
3) We propose to create an automatic reputation evaluation authentication system to take the place of the current email-based system in order to improve the efficiency of authentication.
4) According to the different features of the four retrieval models, we are studying the conditions within which each model can be mostly employed.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: Addison-Wesley, 1999.

[2] D. Bhagwat and N. Polyzotis, "Searching a file system using inferred semantic links," in *HYPERTEXT '05*, Salzburg, 2005, pp. 85-87.

[3] E. Chang, T. S. Dillon and F. K. Hussain, *Trust and Reputation for Service Oriented Environments-Technologies for Building Business Intelligence and Consumer Confidence*: John Wiley & Sons, 2005.

[4] W. Hu, G.-P. Liu, D. Rees, and Y. Qiao, "Design and implementation of Web-based control laboratory for test rigs in geographically diverse locations," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2343-2354, 2008.

[5] A. Leva and F. Donida, "Multifunctional remote laboratory for education in automatic control: the Crautolab experience," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2376-2385, 2008.

[6] E. Chang and M. West, "Digital Ecosystem - A next generation of the collaborative environment," keynotes in *iiWAS2006*, Yogyakarta, 2006.

[7] R. H. L. Chiang, C. E. H. Chua and V. C. Storey, "A smart web query method for semantic retrieval of web data," *Data & Knowledge Engineering*, vol. 38, pp. 63-84, 2001.

[8] C. Dichev and D. Dicheva, "View-based semantic search and browsing," in *WI '06, Hong Kong*, 2006, pp. 919-925.

[9] P. Dini, "Structure and outlook of digital ecosystem research," in *IEEE DEST 2007*, Cairns, 2007.

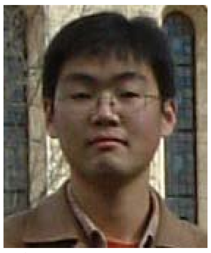
[10] J. Prieto-Blazquez, J. Arnedo-Moreno and J. Herrera-Joancomarti, "An integrated structure for a virtual networking laboratory," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2334-2342, 2008.

[11] S.-C. Wang and Y.-H. Liu, "Software-reconfigurable e-learning platform for power electronics courses," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2416-2424, 2008.

[12] H. Dong, F. K. Hussain and E. Chang, "A transport service ontology-based focused crawler," in *SKG 2008*, Beijing, 2008, pp. 48-55.

[13] A. C. Weaver and M. W. Condry, "Distributing Internet services to the network's edge," *IEEE Transactions on Industrial Electronics*, vol. 50, pp. 404-411, 2003.

[14] J. Gekas, "Web service ranking in service networks," in *the 3rd European Semantic Web Conference (ESWC 2006)*, Budva, 2006.
[15] R. Guha and R. McCool, "TAP: a Semantic Web platform," *Computer Networks*, vol. 42, pp. 557-577, 2003.
[16] W.-P. Lee and T.-C. Tsai, "An interactive agent-based system for concept-based web search," *Expert Systems with Applications*, vol. 24, pp. 365-373, 2003.
[17] D.-R. Liu, M. Shen and C.-T. Liao, "Designing a composite e-service platform with recommendation function," *Computer Standards & Interfaces*, vol. 25, pp. 103-117, 2003.
[18] A. Corallo, G. Passiante and A. Prencipe, *The Digital Business Ecosystem*. Cheltenham Glos Edward Elgar 2007.
[19] I. Toma, D. Roman, D. Fensel, B. Sapkota and J. M. Gomez, "A multi-criteria service ranking approach based on non-functional properties rules evaluation," in *the 6th International Conference on Service Oriented Computing (ICSOC 2007)*, Heidelberg, 2007, pp. 435-441.
[20] H. L. Wang, S. H. Wu, I. C. Wang, C. L. Sung, W. L. Hsu, and W. K. Shih, "Semantic search on Internet tabular information extraction for answering queries," in *CIKM '00*, McLean, 2000, pp. 243-249.
[21] M. Wu, J.-H. She, G.-X. Zeng and Y. Ohyama, "Internet-based teaching and experiment system for control engineering course," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2386-2396, 2008.
[22] A. Sasa, M. z. B. Juric and M. Krisper, "Service-oriented framework for human task support and automation," *IEEE Transactions on Industrial Informatics*, vol. 4, pp. 292-302, 2008.
[23] F. Tao, D. Zhao, Y. Hu and Z. Zhou, "Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system," *IEEE Transactions on Industrial Informatics*, vol. 4, pp. 315-327, 2008.
[24] Y. Takama and S. Hattori, "Mining association rules for adaptive search engine based on RDF technology," *IEEE Transactions on Industrial Electronics*, vol. 54, pp. 790-796, 2007.
[25] J. L. M. Lastra and M. Delamer, "Semantic web services in factory automation: fundamental insights and research roadmap," *IEEE Transactions on Industrial Informatics*, vol. 2, pp. 281-294, 2006.
[26] T. T. Quan, S. C. Hui and A. C. M. Fong, "Automatic fuzzy ontology generation for semantic help-desk support," *IEEE Transactions on Industrial Informatics*, vol. 2, pp. 155-164, 2006.



Hai Dong (S'08) received a Bachelor of Management Degree in Information Management in 2003 from Northeastern University, Shenyang, China. In 2006, he received his Master of Commerce Degree in Information Technology from Curtin University of Technology.
He is currently a doctoral candidate in the Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology. His research interest includes: ontologies, semantic web, information retrieval, digital ecosystems, semantic search, focused crawling, and service computing.

Farookh Khadeer Hussain received a Bachelor of Technology Degree in Computer Science and Computer Engineering. Later, he received his Master's Degree in Information Technology from La Trobe University in Melbourne, Australia. In 2006, he received a PhD in Information Systems from Curtin University of Technology, Perth, Australia.
He is a research fellow in the Digital Ecosystems and Business Intelligence Institute (DEBII). His areas of active research are trust, reputation, trust ontologies, data modeling of public and private trust data. He works actively in the domain of making informed business decisions (business intelligence) through the use of trust and reputation technology. He is interested in the application of trust and reputation as a technology, as a business analysis and intelligence tool, and the applications of trust and reputation to various domains. His other areas of research interests are in Web 2.0, social networks and mashup.

Elizabeth Chang (M'02-SM'07) received a Bachelor's Degree in Computer Science in 1985 from Beijing University, Beijing, China. Later, she received her Master's Degree in Computer Science in 1991 from La Trobe University, Melbourne, Australia. Then she received a PhD in Computer Science in 1996 from La Trobe University.
Since 2003, she has been a professor in School of Information Systems, Curtin University of Technology. Since 2006, she has been the founder and director of Digital Ecosystems and Business Intelligence Institute. She has been awarded the Vice Chancellor's Outstanding Performance Award for 2005 and the Dean's Best Researcher of Year Award for 2005 and 2004. She has co-authored 3 books and has published over 350 scientific papers as book chapters, in international journals and at refereed conferences as well as numerous invited Keynote papers and Tutorials. These also include several ACM and IEEE Transactions papers which are the top journals in the field. She is currently holding 6 ARC large Discovery and Linkage grants and a Tier 1 Centre of Excellence grant and obtained cash from ARC, industry partners as well as Research Centre of Excellence funds of over $5 million for 2002-2011. All her research and development work has been in the areas of IT Applications for Business. She has supervised or co-supervised 15 PhD graduates to completion in the last 6 years. Her research interest includes: ontology and multi-agent systems, data mining for business intelligence, trust, security and risk in e-Business, XML, Web Services, P2P for collaborative environments, web engineering, IT for business and commerce, IT for health informatics, and IT for education.

Appendix

APPENDIX A

TABLE I

TESTING RESULTS FOR ECBR MODEL

Threshold value	Precision	Mean Average Precision	Recall	Harmonic Mean	Fallout
>0	12.38%	70.97%	75.26%	21.27%	12.74%
>0.05	12.38%	70.97%	75.26%	21.27%	12.74%
>0.1	12.38%	70.97%	75.26%	21.27%	12.74%
>0.15	14.86%	70.97%	75.25%	24.83%	11.85%
>0.2	17.21%	71.22%	74.59%	27.97%	9.28%
>0.25	24.75%	71.82%	73.80%	37.07%	5.77%
>0.3	24.75%	71.82%	73.80%	37.07%	5.77%
>0.35	28.15%	74.91%	67.71%	39.77%	3.95%
>0.4	28.01%	78.18%	65.80%	39.29%	3.88%
>0.45	27.95%	78.39%	65.55%	39.19%	3.88%
>0.5	27.95%	78.39%	65.55%	39.19%	3.88%
>0.55	66.46%	90.65%	41.43%	51.04%	0.48%
>0.6	66.46%	90.65%	41.43%	51.04%	0.48%
>0.65	66.51%	90.79%	41.09%	50.80%	0.48%
>0.7	74.41%	91.15%	37.46%	49.83%	0.23%
>0.75	79.43%	90.65%	36.62%	50.13%	0.16%
>0.8	79.43%	90.65%	36.62%	50.13%	0.16%

APPENDIX B

TABLE II

TESTING RESULTS FOR VSM MODEL

Threshold value	Precision	Mean Average Precision	Recall	Harmonic Mean	Fallout
>0	23.13%	67.65%	66.31%	34.30%	5.41%
>0.05	23.35%	67.65%	66.30%	34.54%	5.11%
>0.1	23.90%	68.44%	65.21%	34.98%	4.66%
>0.15	25.32%	69.68%	61.92%	35.94%	3.93%
>0.2	28.33%	71.41%	58.54%	38.18%	3.17%
>0.25	32.36%	73.43%	54.35%	40.57%	2.52%
>0.3	37.36%	74.67%	51.02%	43.13%	1.90%
>0.35	41.57%	76.40%	46.17%	43.75%	1.39%
>0.4	48.35%	78.78%	42.59%	45.29%	1.00%
>0.45	53.08%	81.75%	37.56%	43.99%	0.71%
>0.5	63.93%	86.17%	32.74%	43.30%	0.44%
>0.55	67.09%	85.88%	29.51%	40.99%	0.30%
>0.6	76.65%	87.07%	27.00%	39.93%	0.18%
>0.65	79.02%	84.44%	22.90%	35.50%	0.13%
>0.7	76.32%	81.90%	17.62%	28.64%	0.10%
>0.75	84.00%	89.34%	15.31%	25.90%	0.06%
>0.8	85.71%	87.14%	9.83%	17.64%	0.03%

APPENDIX C
TABLE III
TESTING RESULTS FOR LSI MODEL

Threshold value	Precision	Mean Average Precision	Recall	Harmonic Mean	Fallout
>0	4.13%	58.56%	80.68%	7.86%	45.70%
>0.05	14.29%	62.86%	73.29%	23.91%	9.60%
>0.1	21.08%	65.06%	69.49%	32.35%	5.55%
>0.15	25.04%	66.91%	64.78%	36.12%	4.19%
>0.2	27.96%	68.80%	62.11%	38.57%	3.40%
>0.25	31.35%	71.35%	59.13%	40.97%	2.84%
>0.3	32.82%	72.34%	56.71%	41.57%	2.43%
>0.35	34.72%	73.93%	53.95%	42.25%	2.06%
>0.4	36.15%	75.23%	51.50%	42.48%	1.73%
>0.45	37.79%	77.97%	47.91%	42.25%	1.42%
>0.5	40.09%	79.33%	45.38%	42.57%	1.16%
>0.55	43.74%	80.23%	42.31%	43.01%	0.91%
>0.6	46.76%	80.44%	39.58%	42.87%	0.71%
>0.65	49.48%	80.65%	35.80%	41.54%	0.56%
>0.7	55.19%	81.82%	31.87%	40.40%	0.42%
>0.75	59.95%	82.08%	28.36%	38.50%	0.31%
>0.8	73.06%	83.97%	24.97%	37.22%	0.17%

APPENDIX D
TABLE IV
TESTING RESULTS FOR PROBABILISTIC MODEL

Threshold value	Precision	Mean Average Precision	Recall	Harmonic Mean	Fallout
>0	24.01%	64.23%	64.86%	35.05%	5.01%
>0.05	24.17%	64.51%	63.92%	35.07%	4.71%
>0.1	25.08%	65.27%	63.13%	35.89%	4.11%
>0.15	27.14%	66.87%	61.06%	37.57%	3.47%
>0.2	27.17%	67.99%	63.30%	38.02%	3.64%
>0.25	32.08%	70.32%	55.48%	40.65%	2.38%
>0.3	35.08%	70.76%	52.33%	42.00%	1.92%
>0.35	38.00%	71.62%	46.82%	41.95%	1.36%
>0.4	43.31%	72.78%	45.30%	44.28%	1.15%
>0.45	46.33%	73.21%	44.44%	45.37%	0.94%
>0.5	47.91%	74.48%	39.01%	43.00%	0.74%
>0.55	53.05%	73.37%	36.12%	42.98%	0.52%
>0.6	56.66%	74.89%	35.39%	43.57%	0.47%
>0.65	56.07%	74.60%	33.71%	42.11%	0.43%
>0.7	59.17%	75.09%	28.14%	38.14%	0.33%
>0.75	61.28%	76.30%	25.17%	35.69%	0.28%
>0.8	59.64%	74.99%	23.95%	34.17%	0.28%

APPENDIX E
TABLE V
COMPARISON RESULTS AMONG FOUR MODELS

Model	Optimal Threshold Value	Precision	Mean Average Precision	Recall	Harmonic Mean	Fallout Rate
ECBR	>0.55/0.6	66.46%	90.65%	41.43%	51.04%	0.48%
VSM	>0.4	48.35%	78.78%	42.59%	45.29%	1.00%
LSI	>0.55	43.74%	80.23%	42.31%	43.01%	0.91%
PM	>0.45	46.33%	73.21%	44.44%	45.37%	0.94%

PM stands for the probabilistic model