www.ijiarec.com

ISSN:2348-2079

Volume-6 Issue-1

International Journal of Intellectual Advancements and Research in Engineering Computations

Semantic technologies centered with web data rescue and service collection systems– an analysis

[1]M.Ramu, [2]M.Beemajan Shaheen

[1]Asst Prof & Head, MCA/SVCET, Puliyangudi

[2]Asst. Prof/CSE, SVCET, Puliyangudi

ABSTRACT

Semantic web services focuses on providing automation and dynamics to existing web service technologies. A large amount of effort and money has been invested in the field of semantic web service discovery. In traditional web service discovery techniques, users need to select relevant web services manually from an extensive textual list. The most notable approaches rely on the description of Web services using semantics. This new breed of Web services, termed Semantic Web Services (SWSs) aim to improve the possibilities for automated discovery, composition and invocation of Web Services by providing ontology-based service descriptions expressed in a formal language. Several approaches have been driving the development of Semantic Web Service frameworks such as OWL-S (Ontology Web Language for Services), WSMO (Web Service Modeling Ontology) and IRS (Internet Reasoning Service). This paper focuses on survey on semantic technology centered on rescue of data and service collection system.

Keywords: SWS, WSMO, IRS, Fuzzy matchmaking framework.

INTRODUCTION

As a support for developing complex web based informative and service based systems, the web technology has developed a lot. The emergence of service-oriented computing is the key-enabler of web applications which results in subsequent increase in the web services available [1]. The utilization of syntactic structures like XML has empowered the representation and publication of machine-readable details that might be acquired and utilized by developers and applications. However there are still atleast two major obstacles for enabling and automating the construction of large-scale workflows within open environments: semantic and schematic heterogeneity [2]. Semantic Web Services [3, 4] address this problem by providing a declarative, ontological framework for describing services, messages, and concepts in a machine-readable format that can also facilitate logical reasoning. Thus, service descriptions can be interpreted based on their meanings, provided that there is support for reasoning over a Semantic Web Service description, workflows and service compositions can be constructed based on the semantic similarity of the used concepts.

Semantic Web Services enables the updating of existing Web service descriptions with semantically rich descriptions (annotation), which can be used by applications and middleware to discover, compose, and validate services and workflows. In this paper we provide a survey on various semantic technologies based web data and service systems. This paper provides survey on semantic based systems under two broad categories: Service Collection systems and Data Rescue systems. The rest of the paper is organized as follows; section II provides details about few semantic based service collection systems, section III provides details about few semantic based data rescue systems and

Author for correspondence:
Asst Prof & Head, MCA/SVCET, Puliyangudi
Email: ramu.muthuswamy@gmail.com

section IV provides conclusion for this survey which is followed by references.

Service Collection Systems

This section provides the data about the service collection systems. In [5], Climate change has led to studies focused on the changes in the multinational Arctic region. To facilitate Arctic research, a spatial data infrastructure (SDI), where Arctic data, data, and services are shared and integrated in a seamless manner. Semantic based web service for arctic spatial data are discovered through,

- A hybrid approach for efficient service discovery from distributed web catalogs and the dynamic Internet.
- A domain knowledge base to model the latent semantic relationships among scientific data and services.
- An intelligent logic reasoning mechanism for (semi-)automatic service collection and chaining.

Arctic SD

The ASDI provides an automatic discovery mechanism to collect distributed data, the buildup of hydrology ontology to model the latent semantic relationship among the data, and a smart search and integration service to chain and visualize the datasets to enable a semiautomatic science workflow.

A hybrid approach for efficient service discovery from distributed web catalogs and the dynamic Internet

An automated service in the SDI discovery identifies all the distributed Arctic data resources across a public network and places them in a virtual data repository. The repository is “virtual”.it does not actually include real data, instead only met a data and online address of the real datasets are stored.

A domain knowledge base to model the latent semantic relationships among scientific data and services

The rich resources have been collected and stored in the ASDI repository; there is a need to provide a mechanism to help system find the most suitable data to perform automated analyses for its discovery. The task included,

- Developed a domain knowledge base (ontology) to disambiguate between different terminologies by explicitly defining a conceptualization,
- Provided an intelligent search tool for resource collection in semantic-enabled service chaining.

An intelligent logic reasoning mechanism for (semi-)automatic service collection and chaining

Semantic reasoning is the core component of the semantic search engine. The automatic service collection performed using, syntax analysis, semantic analysis, and data rescue tasks from a heterogeneous environment are performed in sequence. Syntax analysis focuses on analyzing components of a query sentence. The inference engine adopted was the Jena Semantic Web Framework for Java. The processing procedures are as follows.

- Ontology is loaded into the memory or persistent storage maintained by Jena.
- The DL-based queries are transformed into formal SPARQL (Prud’hommeaux and Seaborne, 2005) queries.
- Through the Jena query API, the sub queries are conducted and results are retrieved.
- Query results are combined to obtain expanded and more specific data.

This scheme performed three basic tasks of Service discovery, knowledge base development, and service decomposition and chaining when building an integrated ASDI. The hybrid approach, which combines multi- catalogue searching, and active crawling mechanisms, helps to collect rich resources to support scientific modelling. The provision of diverse resources from the central access point ASDI improves the automatic discovery of spatial data.

In [6], the core semantic search model is based on an adaptation of the classic keyword-based IR model. It spans the four main processes of an IR system: document indexing, query processing, searching and ranking.

Indexing

The system takes as input a user’s natural language (NL) query. Data available in standard

Web pages (the document base) is indexed using semantic knowledge found in the Semantic Web. Indexing is achieved by linking the semantic space to the unstructured content space through explicit annotation of documents with semantic data. Annotation by NLP (Natural Language Processing), using Wraetlic NLP tools, the annotation module analyzes the textual documents. The comparisons are done by using an ontology index created with Lucene, and according to fuzzy metrics based on the Levenshtein distance. Most similar semantic concepts are chosen and added as annotations of the document. After the creation of annotations, a TF-IDF technique computes and assigns weights in the semantic entity index. Annotation can also be based on contextual semantic data.

Query processing

The Natural Language query is processed by an ontology-based Question Answering (QA) system, PowerAqua. PowerAqua is able to answer queries by locating and integrating data, which can be massively distributed across heterogeneous semantic resources.

Searching and ranking

When relevant ontological knowledge is returned as an answer to the user's query, the system retrieves and ranks documents containing data. The document rescue and ranking algorithm is based on an adaptation of the traditional vector space IR model,

Providing fast access to SW data: a SW gateway, WebCORE, the SW gateway that collects, analyzes and gives access to online available semantic content, enabling the experimentation of the rescue algorithms on large amounts of semantic content. WebCORE pre-processes and stores the gathered data in several inverted lexical ontology and/or taxonomical ontology indices.

SW gateway provides a multi-ontology accessing module that allows managing several ontologies at a time within the application. In the Multi-ontology access module, at the first level an OntologyPlugin API is defined.

In the second layer, two implementations of the above API are provided for two SW frameworks: Sesame and Jena Ontologies are added to WebCORE by providing the following data: the ontology identifier, its language, its corresponding framework, and its location. The system outputs a set of ontology elements that answer the user's question and a complementary list of semantically ranked relevant documents.

The system attempts to bridge the gap between the Data Rescue and the Semantic Web communities in the understanding and realization of semantic search. The integration of an external NL query processing module, PowerAqua provides ease of use and heterogeneity, exploiting PowerAqua's ability to answer queries using large amounts of heterogeneous semantic content. The flexible and scalable annotation algorithms generate annotations between large amounts of documents and semantic metadata. The SW gateway provides fast access for applications to SW content.

In [7], an accurate measuring of semantic similarity between two words (or entities) can measured by an empirical method to estimate semantic similarity using *page counts* and *text snippets*. This system provided an automatic method to estimate the semantic similarity between words or entities using web search engines. It is time consuming to analyze each document separately. Web search engines provide an efficient interface to this vast data by using *page counts* and *snippets*. They are two useful data sources provided by most web search engines.

Page count

Page count analysis ignores the position of a wording a page. Therefore, even though two words appear in a page, they might not be actually related. Page count of a polysemous word (a word with multiple senses) might contain a combination of all its meaning.

Snippets

Snippets is a brief window of text extracted by a search engine around the query term in a document, provide useful data regarding the local context of the query term snippets is also efficient because it avoids the trouble of downloading web pages. This system consist of following modules,

- Page Count-Based Co-Occurrence Measures
- Lexical Pattern Extraction
- Lexical Pattern Clustering
- Measuring Semantic Similarity
- Training

Page Count-Based Co-Occurrence Measures

Page count based co-occurrence measures using Jaccard, Overlap (Simpson), Dice, and Point wise mutual data (PMI), to compute semantic similarity using page counts.

Lexical Pattern Extraction

Lexical pattern extraction performed by using snippets. Snippets contain a window of text selected from a document that includes the queried words. Snippets decide whether a particular search result is relevant, without even opening the url. Snippet is not limited to extracting patterns only from the mid fix. Moreover, the consideration of gaps enables us to capture relations between distant words in a snippet.

Lexical Pattern Clustering

This system represent a pattern a by a vector a of word-pair frequencies. It designate a, the word-pair frequency vector of pattern a. It is analogous to the document frequency vector of a word, as used in data rescue. Sequential clustering algorithm used to efficiently cluster the extracted patterns. Clustering algorithm attempts to identify the lexical patterns that are similar to each other more than a given threshold value.

Measuring Semantic Similarity

The co-occurrence measures using page counts are utilized for measuring semantic similarity.

- WebJaccard,
- WebOverlap,
- WebDice,
- WebPMI.

Training

Training data set S is generated automatically from WordNet synsets. It is a manually created English dictionary, to generate the training data required by the system. The two-class support vector machine (SVM) trained to classify synonymous and non-synonymous word pairs.

This system performed a semantic similarity measure using both page counts and snippets retrieved from a web search engine for two words. Four word co-occurrence measure were introduced using page counts. The extraction algorithm to extract numerous semantic relations that exist between two words are utilized. A two-class SVM was trained using those features extracted for synonymous and non-synonymous word pairs selected from

WordNet synsets

In [8], a critical step in the process of reusing the existing WSDL specified services for building large scale distributed and heterogeneous applications is the discovery of potentially relevant services. Semantic Web Services (SWS), augmenting Web service descriptions using Semantic Web technology, were introduced to address the above problem and to facilitate the autonomous publication, discovery, and execution of Web services at the semantic level. Moreover, semantic Web service description languages, such as Ontology Web Language for Services (OWL-S) and Web Service Modeling Ontology (WSMO), were proposed as abstractions of syntactic Web service description languages such as WSDL.

This paper, proposes a fuzzy logic based matchmaking approach to support discovery of potentially relevant Semantic Web Services for the collaborative material collection system. This approach expands the above-mentioned fuzzy matchmaking frameworks by defining the fuzzy set and similarity relation on web service attributes. It is composed of a theoretical framework for fuzzy matchmaking, and a semantic annotation specification of how the needed data of web service attributes can be captured as semantic annotation for WSDL interfaces, operations, faults, and XML Schema, and a fuzzy matchmaking algorithm for calculating the fuzzy similarity degree of web services providing the material collection services in the area of collaborative manufacturing.

Fuzzy matchmaking framework for collaborative material collection services

In collaborative material collection system, capabilities of manufacturers and material attributes are registered as entries to a central semantic service registry provided in a fuzzy matchmaking framework (FMF) for semantic web service execution environment.

FMF is an extended version of the standard Web Services model and Web Services Execution Environment established in the literature, and is also proposed for the development of collaborative material collection system.

FMF is composed of

- Web Service Deploying Module,
- Web Service Registering Module and
- Web Service Calling Module.

Four tools are used in the above deploying and registering process of the service, they are

1. **Web service register tool** reads WSDL documents and semantic model descriptions from the service provider's workspace, and enrolls the web service to the registry.
2. **OWL-to-service implementation tool** encodes the semantic description into the service implementation.
3. **Web service annotation tool** adds semantic data on WSDL document of web service.
4. **Web service deployment tool** compiles the service implementation and deploys it onto the web service server.

Web Service Calling Module, FMF searches, matches and calls the available web service when a company decides a task to outsource through web services; and service requester matches, ranks, selects and executes web services.
This module consists of the following components:

- Service fuzzy matching tool matches, ranks and selects web services by sending a query to the service register center;
- Semantic reader helps the service user with browsing the service semantics;
- WSDL reader enables service caller to understand documents written with WSDL and stored in service registry;
- Service caller calls one of the operations of a chosen web service.

Fuzzy material database design for goal service

The terminological relations in language aspect are from ARTEMIS (Analysis and Reconciliation Tool Environment for Multiple Data System), the semantic relations in context aspect are defined in HELIOS (Helios Evolving Interaction-based Ontology knowledge Sharing) to represent the connection of two concepts.

Semantic annotation for web service attributes

- The semantic data of a web service can be annotated into WSDL by Semantic Annotations for WSDL and XML Schema (SAWSDL).
- SAWSDL provides a standard way by which WSDL documents can be connected to semantic descriptions, such as the semantic data provided by OWL-S (OWL for Services) and other Semantic Web Services frameworks.
- The semantic annotation after the WSDL document is annotated by semantic data from OWL model, it is changed into semantic-annotated WSDL document.

Fuzzy matchmaking for Semantic Web Services

In this section, the fuzzy definition for semantic attributes of a material collection service on the entire material ontology, called fuzzy set in extension is given. Based on these, we define the fuzzy similarity for service attributes and service objects, and give the computing formula of fuzzy similarity degree for material collection service.

Fuzzy definition for semantic attributes of web services

The input, output and effect attributes of Web Services in the research area may be defined on two kinds of domains at a fuzzy-set are *symbolic domains* and *numerical domains*.

This paper presents a fuzzy-set based semantic similarity matching algorithm for semantic annotation attributes of web services to improve the precision rate and the recall rate for material collection services. The service attributes similarity and fuzzy similarity for web services were given in the fuzzy matching algorithm. The algorithm can be used to support a fully automated and veracious service discovery process, by distinguishing among the potentially useful and the likely irrelevant services and by ranking them according to their relevance to the developer's query. A collaborative material collection service has been implemented to demonstrate the effectiveness of the proposed approach.

In [9], an automatic ontology-based knowledge extraction from Web documents has been proposed and used to generate personalized biographies. An ontology-based knowledge management system has been proposed to assist engineers in sharing, searching, and managing knowledge.

This system presents an ontology-based fuzzy support agent for ship steering control. The specific study methods include,

- Establishing a remotely control ship steering system on a PC environment,
- Establishing and applying fuzzy steering maneuvering model to control rudder angle based on fuzzy inference mechanism, and
- Identifying the efficiency by applying a real ship steering system to miniature model steering control system.

The proposed fuzzy agent contains **three main mechanisms**,

- The interpretation mechanism of linguistic instruction,
- The self-regulation mechanism, and
- The task performance mechanism.

Ontology-based fuzzy support agent

Ontology model

The structure of the domain ontology adopted in this paper, including

1. The domain layer,
2. The category layer, and
3. The concept layer

The domain layer represents the domain name of an ontology and consists of various categories defined by domain experts. There are several categories, such as "category1, category 2, category 3. . . category k", defined in the category layer.

In the concept layer, each concept contains a concept name Ci, an attribute set {ACi1; . . . ; ACiqig} and an operation set {OCi1; . . . ;OCiqig} for an application domain. Three types of inter-conceptual relations have been used in the domain ontology, namely, the generalization, the aggregation and the association.

Fuzzy support agent

The proposed ontology-based fuzzy support agent consists of a linguistic instruction ontology and a fuzzy support agent. The fuzzy support agent is composed of three mechanisms, including

- The interpretation mechanism,
- The self-regulation mechanism, and
- The task performance mechanism.

Linguistic instruction ontology

The linguistic instruction ontology stores and represents the background knowledge of the linguistic instruction module and the dialogue module. LHM means a pre-defined linguistic hedge module. The linguistic instruction method remotely controls the steering gear.

An ontology-based fuzzy support agent for ship steering control has been performed using the fuzzy support agent is presented to build the manoeuvring models of steersman and the miniature model for steering control system. This system performed a remotely controlled ship manoeuvring and job management and it reduce burden of the ship operators would be greatly alleviated when it comes to the situation where the ship operators observe the exterior conditions by the eye, while reading the data from numerous navigation equipment.

In [11], Service-oriented computing (SOC) is a distributed computing paradigm that makes use of services as the fundamental building block of distributed systems.

The open nature of SOC causes two major challenges considering the service matchmaking approaches:

1. The first challenge arises due to development of services by independent parties. A service matchmaking approach should be able to identify such situations.
2. Second challenge arises due to loosely coupled structure of services. A desired service matchmaking approach should handle such situations and find services that can partially satisfy the request of the service consumers.

The current service discovery mechanism of WSs is based on WSDL and UDDI. The combination of UDDI with WSDL provides a basic mechanism for service discovery, it lacks support for automated discovery. Therefore, SOC needs an automated service discovery approach that will take in a service request and find all the services that can fully or partially satisfy the request.

Proposals to automate service matchmaking

Idea of interface matching

This approach is based on defining the requested service through its expected input–output interface and comparing this expected interface with the input–output interfaces of available services to find matching services.

Advantage

A major advantage of this approach is its simplicity.

Disadvantage

This approach fails to be successful in many situations since it relies only on syntactic knowledge: Two services with identical input–output interfaces may be performing drastically different computations. Since syntactic knowledge does not capture any meaning, matching solely based on syntactic knowledge is not enough.

- In order to overcome this problem, interface matching approach is further improved by the WS community from the syntactic level to semantic level by associating the elements of the interfaces with concepts from ontologies. With this approach one can reason about their semantic similarity and reach a conclusion based on that, even if the input and outputs are not identical.

Advantage

This approach improves the quality of matchmaking results since it does not depend on the syntactic keywords but semantic concepts.

Disadvantage:

Although introduction of semantic knowledge improves the performance of interface matchmaking, it is not possible to determine 'what a service does?' only by considering its interface, this approach may still mistakenly match many services to a service request.

- A possible improvement to overcome such a problem is the use of the data about precondition and effects. Here the service explicitly defines what the precondition and the effect of the service is. To go beyond interface or precondition-effect matching, it is necessary to understand what a service consumer expects from a service as well as what a service can accomplish for a consumer. Only by understanding and using these data, we can achieve better matchmaking performance. This understanding can only be possible if we have knowledge about the business processes of services.

Advantage

Explicit definition of the precondition and the effect of service.

Disadvantage

A service's process need not specify all the internal properties of the service. From a consumer's point of view, it is important to know how a service uses the given input to produce the output, whether any constraints are being enforced, and so on.

To achieve well-targeted matchmaking between service requests and service offerings, service providers need to specify their business processes and service consumers need to specify their requests with well-defined semantics. For this purpose, we use **model checking techniques** in our approach. The main idea of our approach is to transform the service matchmaking into model checking in which we represent services as system models and service request as set of formal properties.

By using model checking as a reasoning mechanism, we determine whether a service can satisfy required properties of a service consumer and accordingly we find matching services to service consumer requests.

Idea for developing a family of methods and algorithms for matchmaking

1. Our first method provides only exact matching capabilities. However, in many settings, finding partially matching services is also important. For this reason, we develop two more methods.
2. Our second method benefits from using semantic concepts. In this method, if we cannot find an exact match, we generate similar requests to the original request using an ontology and the properties involved in the original request. After that we compute a similarity value between the original and generated requests using a semantic similarity metric to capture the degree of match.
3. Our third method is based on the idea of relaxing service consumer requests. In this method, similar to the other method to handle partial matches, we generate similar requests, when there is no exact matching service for a request.

However, compared to the previous case, this time we revise the temporal constraints of the request instead of the semantic properties.

Advantages compared to other approaches such as the input-output precondition-effect matching

- The first advantage is that it captures the business model of a service and hence provides more precise details to the user about the workings of the services.
- The second advantage of our approach is the involvement of the temporal requirements of the users in the service request. This enables the user to phrase precise requirements about the service. Such requirements of the users are not involved with either of the other approaches.

In this paper they present a novel service matchmaking approach that is based on model checking. In this approach they represent services as system models and service requests as a set of LTL formula, in which each formula corresponds to a specific functional property required by the service consumer. The service matchmaking approach is capable of handling both exact and partial matching of services for service requests. In order to capture partially matching services they propose two methods, which are based on user request restructuring. The first method uses ontologies to capture semantic similarities between concepts that are involved in service models and service requests and accordingly restructures service requests. Our second method uses entailment and subsumption relations between LTL operators through a hierarchical structure and uses these relations for service request restructuring.

In [12], with increasing demand on composite Web services, a natural language interface to Web services, which can be used even by a novice user has been posed. Given a user's natural language request to a composite service, a sophisticated abstract workflow from complex sentences with phrases and control constructs is constructed from services that are described and published based on ontologies.

Services are described in a service registry using OWL-S, which is a de facto standard ontology for describing the semantics of Web services. Additionally, the action, object, and parameters of a service are associated with appropriate concepts of the proposed ontologies along with the Input/output ontology.

To extract an abstract workflow from a natural language query, the RASP system, a natural language processor is used. A request is divided into sentence blocks based on control constructs and verbs. Workflow templates are extracted by applying basic workflow patterns to control constructs. In the case of a user's request with a complicated control flow, alternative templates are available from which the best match is determined by considering the semantics and structure of sentence blocks.

For each sentence block, it determines its service type and a list of candidate services from which a service is selected by calculating a similarity with the sentence block and generates an abstract workflow by combining the workflow templates and a more specific service is selected by considering the ancestor-descendant relation between the concepts mapped. Specifically, the natural language representations of the I/O parameter concepts of a candidate service are compared with a sentence block. There are limitations in handling natural language requests including negative sentences. The ambiguity of natural language expressions also causes different interpretations from a user's intention. The method has to be enhanced to deal with a user's request with a number of control constructs more efficiently.

An efficient method that extracts an abstract workflow, which can be used to construct an actual composite service, from a user's natural language request and also deals with a request with a complicated control flow.

In [16], a new customizable and effective matchmaker, called SAWSDL-iMatcher supports several kinds of matching strategies for discovering proper services.

SAWSDL service definition

SAWSDL, designed as an extension of WSDL enriches the service description with two kinds of attributes: model reference and schema mapping. The value of model reference is considered to be used in automated service discovery and composition, while the value of schema mapping is used when mediation code is generated to support invocation of a Web service.

Matching strategies in SAWSDL-iMatcher

A semantic matching strategy for semantic annotations of service operations is proposed and several syntactic matching strategies for each type of service description and statistical-model-based matching strategies are built.

Syntactic matching strategies

Name-based matching strategies

Several name-based matching strategies are built in SAWSDL-iMatcher by exploiting the string similarity measures implemented in Simpack.

Description-text-based matching strategies

The description text that consists of comments written in natural language by service Developers is represented by the classic vector space model: term frequency-inverse document frequency (tf-idf) model. The description texts are preprocessed, using a tokenizer, stemmed with the Porter stemmer, and filtered with a list of stop-words. The description text of SAWSDL document is represented as a vector. SAWSDL-iMatcher supports seven vector-based similarity measures (including cosine, dice, Euclidean, Jaccard, Manhattan, overlap, and Pearson's correlation coefficient) from Simpack for comparing the similarity of description text.

Semantic-annotation-based matching strategies

The similarity between semantic annotations can be measured by syntactically comparing the sets of semantic concepts. The degrees of semantic matching on semantic annotations are determined by the subsumption relationships in domain ontologies. SAWSDL-iMatcher supports a relaxed semantic matching strategy.

Statistical-model-based matching strategies

SAWSDL-iMatcher supports several statistical model-based matching strategies by using different algorithms from Weka, such as simple linear regression, J48 decision tree, logistic regression, support vector regression (_-SVR), etc., to induce the statistical models.

SAWSDL-iMatcher

SAWSDL-iMatcher exploits Pellet as its semantic reasoner and OWL API7 as the interface for accessing OWL-based domain-specific ontologies. The SAWSDL descriptions in SAWSDL-iMatcher exploit OWL to represent the semantic models. The core of SAWSDL-iMatcher is the iXQuery framework for user interface. iXQuery approach to extends XQuery with similarity joins for SAWSDL service matchmaking.

SAWSDL-iMatcher chooses Saxon as a Java implementation of XQuery to support similarity joins without having to install a XML database. Multiple existing similarity measures library like SimPack, are employed to compose sophisticated user- and data-specific similarity joins.

SAWSDL-iMatcher is a customizable matchmaker, which provides a general framework for Semantic Web Service discovery, together with several effective matching strategies from syntactic, semantic to hybrid aggregated matching strategies

In [17], a semantic data system for the advertisement, rescue and collection of application services, and markets that trade resources, in a democratized Grid e-marketplace environment is presented. The system involves the development of the Grid4All Semantic Data System (G4A-SIS). Grid4All market services and application-specific services are being advertised in this system.

The advertisement and matchmaking processes have semantic annotation of WSDL specifications, whose purpose is to explicate the semantics of WSDL specifications and generate the semantic description of services' signatures. For the semantic annotation of services, the Annotation Tool (WSDL-AT) developed in the context of the Grid4All system exploits the WSDL specification, and domain ontology. It provides mappings of WSDL messages' parameter names to ontology classes with respect to their intended semantics. WSDL-AT is provided as a platform-independent stand-alone tool, with a graphical user interface (GUI). The WSDL and the corresponding textual annotations of services created by the tool are submitted to the G4A-SIS and translated to an OWL-S description using methods provided by the G4A-SIS API. The method that implements the translation (namely the WSDLToOWLS method), uses the Mindswap OWL-S API.

Given the semantic annotations of WSDL input/output parameters, WSDL-AT provides these as input to the WSDL2OWL-S method, which finally constructs the services' signature in OWL-S. This profile is registered to the G4A-SIS.

G4A-SIS integrates an advertisement (registration) and a rescue (matchmaking) module for application services. A natural way to rank providers is by considering them in a consumer-centric fashion. G4A-SIS is implemented using Java Enterprise technologies. In addition, Jena provides classes and interfaces that correspond to all the concepts of RDF(S), OWL languages and uses a simple SPARQL engine, ARQ, through which SPARQL queries are executed during the matchmaking process. Also, Jena offers the capability of attaching inference engines, such as Pellet, to the models (ontologies).

The G4A-SIS interacts with a MySQL database via Jena, for the persistent storage of RDF/OWL models (optionally). Automatic classification in matchmaking mechanism for locating market/application-specific services, is actually executed when objects are registered, and not during query processing. The implemented system has been tested on the Apache Tomcat server. The AXIS 1.4 Web services framework is used for enabling web service access for the G4A-SIS API. The G4A-SIS API, supported by AXIS, conforms to WSDL 1.1 specifications to support overloading of operations of the API.

The Collection module is a cross-platform service which is implemented using Java. This module has been tested in standalone mode using the JUnit testing framework. The G4A-SIS provides a Java API that has been developed to support agents to use the G4A-SIS as a web service. The JavaBeans classes are wrappers of corresponding classes in the G4A SIS ontology.

G4A-SIS is implemented as a centralized system, supporting the registration of markets and application-specific services in a central semantic registry. The system has been designed considering the P2P architecture and a modular version of the G4A-ontology to overcome the single point of failure for Grid4All infrastructure. Also, G4A-SIS is dependent on OWL and OWL-S standards. The strong point of the G4A-SIS is in satisfying agents preferences while ensuring high levels of query load balance.

In [19], The SOA of syntactic discovery only enables to produces coarse irrelevant results or sometimes no results. Semantic discovery process in SOA to enable relevant and a new architecture of SOA has been proposed.

Semantic SOA Incorporates a new adaptive technique called social learning that improves service provider’s domain ontology from service consumer’s concept contributions and thus makes the service more semantically discoverable. In SOA, a service provider exposes its services in standard Web Service Description Language (WSDL).WSDL specification the service consumer can make software clients to consume the service.

The UDDI gives API for searching web services for the service requester.

The Improved Architecture of SOA for Semantic Discovery of Web Services

SOA consists of three entities:

- Service provider,
- Service requester
- UDDI.

The semantic SOA enhanced with two actors,

Alignmentor

The task of the alignmentor is to match ontologies with specified algorithm and send the threshold value to the UDDI. It also merges the ontology and send the result to Ontology Handler.

Ontology Handler

The main task of Ontology Handler is to perform CRUD (create, read, update and delete) operation in ontology URI.

This system architecture introduced with two new algorithms for Semantic Matching Process of Web Service Descriptions and Social Learning on Web Service Descriptions through Concept Alignment.

- Semantic matching process algorithm performed for semantic similarity measures for weighted ontology.
- A new algorithm for social learning expressed with applying merging algorithm on weighted ontology.

The implementation tools contain a private JUDDI server, Protégé ontology creator and validator and Jena APIs for ontology matching and alignment.

A simulation test bed is created with a real-world OWL-S service rescue test collection OWLS-TC v4 to evaluate different aspects of the architecture. This collection contains services which are retrieved mainly from public IBM UDDI registries, and are semi-automatically transformed from WSDL to OWL-S. More specifically, it comprises a set of ontologies

The new architecture of SOA is proposed which enables semantic discovery of web services with an adaptive learning technique termed as social learning which produces more relevant discovery results over times. Through application of matching and merging processes of the semantic architecture semantic meaning is preserved after merging. It reduce the noisy results of existing syntactic discovery processes significantly in social learning

Data rescue systems

This section provides the data about the data rescue systems. In [10], this paper presents, Sem-Fit, a semantic hotel recommendation expert system, based on the consumer's experience about the recommendation provided by the system. The proposed expert system uses the consumer's experience point of view in order to apply fuzzy logic techniques to relating customer and hotel characteristics.

Hotel characteristics are represented by means of domain ontology. After receiving a recommendation, the customer provides a valuation about the recommendation generated by the system. Based on the customer's valuations, the rules of the system are updated in order to adjust the new recommendations to the past user experiences.

Capturing the fuzzy knowledge about hotel recommendation

The main elements in our proposal about capturing knowledge in hotel recommendations:

1. The semantic descriptions of the hotels.
2. The fuzzy relationship between the characteristics of the customers and the characteristics of the hotels.
3. The customer feeling about the recommendations.

Recommendation process

The main steps of the recommendation process are:

- Obtaining the customer characteristics.
- Fuzzification.
- Once the customer profile has been fuzzified, the fuzzy rules are evaluated obtaining the set of fuzzy values for the hotel characteristics, as well as its tag for defining the level of adequation based on the affect grid.
- The results obtained are defuzzified in order to obtain a set of concrete characteristics for the hotels.
- With the defuzzified results, hotels with the obtained characteristics are retrieved based on the hotels ontology and the annotations.
- Next the fuzzy recommendation system will calculate the weight of each hotel based on the values of the suitability obtained from the decision matrix.

Three layer scheme that represent the recommendation system architecture

- The first layer corresponds to the user interface. There are two different elements in the user interface layer.
  a. The administrator GUI consists of a web-based application to allow the definition of the fuzzy sets as well as the membership function for each fuzzy set.
  b. The GUI recommendation consists of a web questionnaire in which the customer answers a set of questions proposed by the expert.
- The second layer represents the business logic. This layer contains the fuzzy engine and the semantic engine. The fuzzy engine will evaluate the fuzzy rules defined by the expert in order to determine the most suitable hotel characteristics for a given customer profile based on the affect grid.
- Finally, the persistence layer stores the knowledge about the hotel recommendation. As mentioned, on the one hand, the hotel ontology defines the relevant characteristics of each hotel. The concepts of the hotel ontology describe the category of each hotel based on stars, the room characteristics, the special equipment of the hotel, etc.

A Sem-Fit, a semantic hotel recommendation expert system, has been proposed based on consumers' experience about the recommendation provided by the system. The proposed expert system uses the consumers' experience point of view in order to apply fuzzy logic techniques to relate customers and hotels characteristics, represented by means of domain ontologies and affect grids. After receiving a recommendation, the customer provides a valuation about the recommendation generated by the system. Based on the customers' valuations, the rules of the system are updated in order to adjust the new recommendations to the past user experiences. The validation accomplished shows that the sum of star and suggested hotels of the fuzzy systems obtains better results than the expert recommendation. Moreover, the values of precision and recall and F1 reveal that the Sem-Fit recommendations are on the same level as an expert in the domain and the customer will be able to express his feelings in the same way that he does with the expert.

In [13], known and tested technologies can provide some guarantee of robustness and scalability, but the question of maturity and interoperability between many novel approaches spurred by the Semantic Web concept still remains. Hence, a development experiment aimed at using only novel technologies in developing a Semantic Web application has been formulated. The idea of building a Semantic Web search engine involves the interoperability of different technologies, such as crawling the Semantic Web with the use of intelligent agents as well as storing and querying semantic data through a semantic database.

WebOWL is a Semantic Web search engine for OWL data built on the principles of current search engines but focuses on the actual entities within ontologies. It consists of a community of intelligent agents, acting as crawlers, a semantic database and a ranking algorithm. To discover new ontologies and refresh the data of already stored ones, WebOWL uses BioCrawler, an intelligent crawler that learns to recognize and remember sites that contain ontologies or link to other sites containing ontologies, thereby forming a neighborhood of semantic content. The system consists of cooperating intelligent agents running on the Jade platform.

The database of the WebOWL search engine is an enhanced version of db4OWL, a generic OWL database based on the db4o object database engine that uses Jena's parser and reasoner to import data. The WebOWL search engine uses a web front end to allow users to formulate queries and navigate through the results. The frontend was primarily designed as a demo for the search engine's functionality.

OWLRank is an algorithm developed specifically for WebOWL and is used to determine the ranking value of OWL objects by employing simple heuristics and measures semantic links between classes and individuals to determine their significance. It uses a popularity measure to determine the importance of OWL classes and individuals.

An important issue in the experiment is the usability issue since WebOWL is suitable only for expert users and Intranets. The creation of easy to use and intuitive user interfaces for semantic applications

The WebOWL Semantic search engine is a successful proof of concept system demonstrating a ''purist'' Semantic approach. The system aims to explore how emerging and new technologies can work in tandem in the Semantic Web environment

In [14], the system has introduced a rule-based method for learning ontology instances from text that helps domain experts with the ontology population process and has defined a lexico-semantic pattern language that, in addition to the lexical and syntactical data present in lexico-syntactic rules, also makes use of semantic data. A method has been defined to discover new data automatically.

Hermes data extraction language

The Hermes Data Extraction Language (HIEL) employs semantic concepts from domain ontologies. The language is evaluated in the context of extracting events and relations from news, as an extension to the existing Hermes news personalization framework. Hermes is a framework that can be used for building a personalized news service. It is based on GATE and employs lexico-semantic patterns, which use data from an OWL ontology.

The process of developing the ontology is an incremental middle-out approach. An data extraction language is proposed for updating the ontology semi-automatically that can extract new instances of concepts and relations from news items. The language is characterized by supporting syntactic features, orthographic features, and concepts, relations between concepts, logical operators, repetition, and wildcards and has the expressivity of regular expressions.

Employing ontology elements in the rules

By employing ontology elements, semantics are added to the rules. When ontologies are employed in the rules, potentially one rule is used to describe multiple lexical representations.

Hermes data extraction engine

Based on the language defined, the Hermes Data Extraction Engine (HIEE) has been implemented.

Hermes news portal (HNP)

The implementation of the Hermes framework has the Hermes News Portal (HNP), which allows users to formulate queries and execute them on the domain ontology in order to retrieve relevant news items. The HNP application is a stand-alone, Java-based tool which makes use of various Semantic Web technologies. Within the Hermes News Portal, time-specific features are exploited, time functionalities are added to SPARQL, which resulted in tSPARQL. Within HNP, the classification of the news articles is done using GATE and the WordNet semantic lexicon.

A general framework has been developed that supports the Hermes Data Extraction Engine (HIEE) plug-in. Before the rules are employed to match patterns in text, a few processing tasks are performed, like tokenization, sentence splitting, and Part-Of-Speech (POS) tagging, which are dealt with by the GATE architecture. GATE provides a pipeline consisting of different components, each of which handles a different aspect of the language processing. The components that are part of the pipeline, and come with GATE by default, are in order of usage: Document Reset, ANNIE English Tokenizer, ANNIE Gazetteer, ANNIE Sentence Splitter, ANNIE Part-Of-Speech Tagger, and Onto Gazetteer. After preprocessing a news corpus, the Hermes Data Extraction Rule Engine compiles the rules in the Rule Compiler and matches these rules to the text using the Rule Matcher.

Hermes plug-in

To evaluate the usability and expressivity of the proposed data extraction language, the HIEE plug-in for the Hermes framework was created.

The system is based on Hermes Data Extraction Language (HIEL), a lexico-semantic pattern language, is more easy to use for expressing lexico-semantic patterns than the current state-of-the-art JAPE language. Also, it has been proven that the lexico-semantic approaches over lexico-syntactic ones are superior with respect to both precision and recall and employs ontology concepts and relations.

In [15], a complete ontology-based framework for the extraction and rescue of semantic data in soccer domain using keyword-based semantic rescue approach is presented. The system consists of a crawler module, an automated data extraction module, an ontology population module, an inferencing module, and a keyword-based semantic query interface. The main concern is creating a scalable and user-friendly data rescue system with high rescue performance. Scalability concerns are inferencing, assured by dividing the whole logical model into individual independent models and querying, assured by transforming inferred knowledge by a single special inverted structure.

Approach to semantic rescue

A central soccer ontology utilized by every aspect of the system is designed following an iterative development process.

In the crawler module, usable data from websites such as UEFA and SporX are crawled and temporarily stored by the web crawler. The output of the web crawler is some basic data specific to the game and natural language texts which are used as input to our data extraction module.

Data extraction (IE), a template-based IE approach which unlike other automated approaches does not use linguistic tools, adds structured data to the knowledge base by processing unstructured resources in which data crawled from the Web sites are used. IE module has two parts:

- Named entity recognizer for recognizing and tagging the named entities.
- Two level lexical analysis where complex semantic entities and relations are extracted according to pre-defined templates.

Ontology population is the process of knowledge acquisition by transforming or mapping unstructured, semi-structured and structured data into ontology individuals. Having the output of the IE module, the ontology population process becomes creating an OWL individual for each object extracted during IE. Normally, every event in the ontology has its own set of properties. Data is added to the ontology by creating OWL individual for each of the events in the knowledge base.

For the inferencing module, Pellet, an open-source DL-reasoner, which supports all the standard inferencing services such as consistency checking, concept satisfiability, classification and realization is used. To infer more interesting data, Jena rules are used. For data Rescue in semantic knowledge base, semantic indexing (key word-based querying) is used to achieve high rescue performance and scalability. The indexing mechanism is built up on Apache Lucene, an indexer and searcher, which is essentially designed for free text search.

A user-friendly, high performance and scalable semantic rescue system has been formulated. The system answers complex semantic queries without requiring formal queries such as SPARQL. The structural ambiguities can be resolved easily using semantic indexing.

In [18], the system presents a model of semantic annotations for describing the Web services and an algorithm which discovers and composes the Web services. The OWL-S language is adopted for Web services composition and their inter-operability. For solving the Web services Discovery and Composition, the Web services and the request must have the same modelling, which is “an atomic modeling”, in order to facilitate the matching process. Conceptually, representation of a Web service is seen as a black box.

The composition algorithm

To find automatically a composite service which will satisfy the request, a composition algorithm has been formed, which explores the semantic network in depth-first and backward chaining strategy, in a single pass. The main steps in the algorithm are:

1. All services whose outputs that are similar with the goal are searched in the Semantic network.
2. For each service found, a composition plan is created.
3. For each new goal which is in the set of the inputs of the services:
4. All services whose outputs are similar with the new goal are searched.
5. If (all the outputs are treated) or (there are no more goals) then the process is stopped, else returned to step (3).

The backward chaining focuses on the goal avoiding the deduction of useless conclusions. To validate the system, a prototype in java under the operating system Windows XP and the environment JBuilder with free tools and open sources has been implemented. Several Java-API for handling services described in OWL-S are used and also, the Pellet reasoner is used to determine the similarity measure. Pellet has been chosen as the composition requires some kind of reasoning capability: Pellet handles individuals, doesn’t make the unique Name Assumption, supports entailment checks and works with XML schema datatypes indexed by an ontology.

The proposed approach uses an inter-connected network of semantic Web services described in OWL-S, using the similarity measure (outputs–inputs similarity) between concepts based on ontology, built before any submitted request. With a single exploration, the composition algorithm can find several composition plans. The selected composition plan chosen is the “the best one” according to the quality criteria. This technique takes advantages from a graph structure, chaining algorithm of expert system and semantic annotations.

CONCLUSION

This paper shows the survey about the a hybrid system that put together concepts of knowledge and collaborative filtering that allows suggestions to be made based on (1) service collection systems, and (2) data rescue system. This work also introduce the service descriptions can be interpreted based on their meanings, provided that there is support for reasoning over a Semantic Web Service description, workflows and service compositions can be constructed based on the semantic similarity of the used concepts.

REFERENCES

[1]. Konstantinos Tserpes ∗, Fotis Aisopos, Dimosthenis Kyriazis, Theodora Varvarigou "A recommender mechanism for service collection in service-oriented Environments" Future Generation Computer Systems, Volume 28(8), 2012, 1285–1294

[2]. Hai H.Wang, Nick Gibbins, Terry R. Payne, Domenico Redavid, "A formal model of the Semantic Web Service Ontology (WSMO)", Data Systems 37, 2012¸33–60

[3]. S.A. McIlraith, T.C. Son, H. Zeng, Semantic web services, IEEE Intelligent Systems 16(2), 2001, 46–53 doi:doi.org/10.1109/5254.920599.

[4]. T. Payne, O. Lassila, Guest editors' introduction: semantic web services, IEEE Intelligent Systems 19(4), 2004, 14–15 doi:doi.org/10.1109/MIS.2004. 29.

[5]. W. Li, C. Yang, D. Nebert , R. Raskin , P. Houser , H. Wu , Z. Li, "Semantic-based web service discovery and chaining for building an Arctic spatial data infrastructure", Computers and Geoscience, 37, 2011, 1752 – 1762.

[6]. Miriam Fernández, Iván Cantador, Vanesa López, David Vallet, Pablo Castells, Enrico Motta, "Semantically enhanced Data Rescue: An ontology-based approach", Web Semantics: Science, Services and Agents on the World Wide Web 9, 2011, 434–452.

[7]. Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuk, "A Web Search Engine-Based Approach to Measure Semantic Similarity between Words", IEEE Transactions on Knowledge and Data Engineering, 23(7), 2011, 757 – 767.

[8]. Min Liu, Weiming Shen, Qi Hao, Junwei Yan, Li Bai, "A fuzzy matchmaking approach for Semantic Web Services with application to collaborative material collection", Computers in Industry 63, 2012, 193 – 209.

[9]. Ki-Yeol Seo, Gyei-Kark Park, Chang-Shing Lee, Mei-Hui Wang", Ontology-based fuzzy support agent for ship steering control", Expert Systems with Applications 36, 2009, 755–765.

[10]. Ángel García-Crespo, José Luis López-Cuadrado, Ricardo Colomo-Palacios, Israel González-Carrasco, Belén Ruiz-Mezcua, "Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain", Expert Systems with Applications 38, 2011, 13310–1331.

[11]. Akin G, Pinar Yolum, Bogazie, "Service matchmaking revisited: An approach based on model checking", Web Semantics: Science, Services and Agents on the World Wide Web 8, 2010, 292–309.

[12]. JongHyun Lim, Kyong-Ho Lee, "Constructing composite web services from natural language requests", Web Semantics: Science, Services and Agents on the World Wide Web 8, 2010, 1–13.

[13]. Alexandros Batzios , Pericles A. Mitkas, "WebOWL: A Semantic Web search engine development experiment", Expert Systems with Applications 39, 2012, 5052–5060.

[14]. Wouter IJntema, Jordy Sangers, Frederik Hogenboom, Flavius Frasincar, Erasmus University Rotterdam, "A lexico-semantic pattern language for learning ontology instances from text", Web Semantics: Science, Services and Agents on the World Wide Web, 15, 2012, 37 – 50.

[15]. Soner Kara,O¨zg ¨ ur Alan, OrkuntSabuncu, SametAkpınar, NihanK.Cicekli , FerdaN.Alpaslan, "An ontology-based rescue system using semantic indexing", Data Systems, 37(4), 2012, 294 – 305.

[16]. Dengping Wei, Ting Wang, Ji Wang, Abraham Bernstein, "SAWSDL-iMatcher: A customizable and effective Semantic Web Service Matchmaker", Computer Standards & Interfaces 31, 2009, 1108–1117.

[17]. George A. Vouros, Andreas Papasalouros, Konstantinos Tzonas , Alexandros Valarakos , Konstantinos Kotis , Jorge-Arnulfo Quiané-Ruiz , Philippe Lamarre , Patrick Valduriez, "A semantic data system for services and traded resources in Grid e-markets", Future Generation Computer Systems 26, 2010, 916-933.

[18]. Hassina Nacer Talantikite , Djamil Aissani , Nacer Boudjlida, "Semantic annotations for web services discovery and composition", Computer Standards & Interfaces, 31, 2009, 1108–1117

[19]. Sajib Kumar Mistrya,, Mosaddek Hossain Kamalb, Dilip Mistry, "Semantic Discovery of Web Services through Social Learning", Procedia Technology 3, 2012, 167 – 177.

Currently working as a Head, Master of Computer Applications Department in S.Veerasamy Chettiar College of Engineering and Technology, Puliyangudi. Having nearly 9 years of teaching experience with vast knowledge in networking field.	Currently working as Assistant Professor, Computer Science and Engineering Department in S.Veerasamy Chettiar College of Engineering and Technology, Puliyangudi. Having nearly 10 years of teaching experience and having familiarity in networking field.