Knowledge Provision with Intelligent E-Services

Panos Georgolios,† Kostas Kafentzis,‡ Gregoris Mentzas*

Information Management Unit, National Technical University of Athens, 9, Iroon Polytechniou str., 157 80 Zografou, Athens, Greece

Interorganizational knowledge networks and knowledge marketplaces have emerged to enable organizations to share or commercially exploit their knowledge outside narrow organizational borders. The materialization of these structures requires concrete and sound mechanisms for the efficient external provision of knowledge stored in knowledge repositories within the organization. In our approach, we employ semantic Web services as a vehicle for publishing knowledge repositories. We propose “knowledge provision services” as a means for efficient retrieval and composition of knowledge objects from knowledge repositories of various organizational contexts regardless of the environment within which they are delivered. In this direction, we have extended OWL-S with a knowledge object ontology, which represents knowledge objects in a generic, application-neutral way, and we have developed an infrastructure for the publication, discovery, composition, and delivery of Knowledge Provision Services founded on the Web services architecture. © 2007 Wiley Periodicals, Inc.

1. INTRODUCTION

Organizational knowledge is a crucial production factor whose importance has gained extreme recognition during the past decade.¹,² To harvest and leverage organizational knowledge, an increasing number of organizations have undertaken major Knowledge Management (KM) investments, which have resulted in the development of several KM systems and subsequently in the massive accumulation and storage of knowledge produced through organizational operations. These systems provide the underlying technological means that may support a wide set of KM activities, ranging from intelligent information retrieval from knowledge repositories to internal knowledge markets.¹

As knowledge management has matured, another trend has emerged; organizations have become interested in sharing or commercially exploiting their

*Author to whom all correspondence should be addressed: e-mail: gmentzas@mail.ntua.gr.
†e-mail: pgeorgol@softlab.ntua.gr.
‡e-mail: kafe@softlab.ntua.gr.

knowledge outside their narrow organizational borders. Interorganizational knowledge networks with customers, suppliers, and, in some cases, even with competitors and knowledge marketplaces have emerged to harness this opportunity.  

This trend has led to the need for concrete and sound mechanisms for providing knowledge outside the organization as “knowledge provision services” that aim to enable efficient retrieval and composition of knowledge objects from knowledge repositories of various organizational contexts regardless of the environment within which they are delivered, for example, a market or a value chain. The following is a simple example of a knowledge provision service.

Let us consider a new business development manager who aims to set up a new business in China. The manager seeks a solution to this problem. By turning within the internal organizational surroundings, a solution may be found either by a colleague who has relevant knowledge or within guides and best practices in the firm’s knowledge repositories. If the solution cannot be found within the firm, the manager turns to external knowledge sources. The needed knowledge may come in the form of a road map to setting up new businesses in China—such documents are provided by chambers of commerce all over the world, best practices from other firms that have undertaken similar endeavors in the past—such documents are usually provided by specialized consultancies either in document form or as personalized advice, or as a pointer to the contact data of a knowledgeable freelance consultant who has experience in setting up businesses in China.

Hence, what the manager needs is a “knowledge provision service” that will orient her toward the appropriate knowledge objects. These knowledge objects need to be discovered, retrieved, evaluated, selected, and delivered.

When restricted within the organizational borders, the technology used for representing and enriching knowledge objects with metadata that conceptualize the application domain is ontologies. This ontological knowledge can be used to enable assessments of knowledge objects’ content and their potential usage.

However, when moving from the internal surroundings of an organization to the external environment, a number of heterogeneity problems take place that complicate knowledge provision. This is due to the differences in mind-sets, languages, and ontological concepts that people (and systems) adopt when they come from different organizations.

The most suitable environment for technologically supporting knowledge provision services is the use of semantic Web services, whose semantic descriptions allow external agents to understand their functionality and internal structure so the agents can discover, compose, and invoke such services. In this area, researchers have proposed OWL-S (formerly DAML-S), which uses OWL in combination with WSDL and SOAP. OWL-S is a OWL-based Web service ontology that supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of Web services in unambiguous, computer-interpretable form.

The previous example could pinpoint our approach more eloquently. The central question is how the manager of our example could use a knowledge provision service that would retrieve all the necessary knowledge objects and help her deal with the new business development problem that she faces. The attributes of OWL-S
are not enough for a clear declaration of the knowledge objects to be leveraged by the service. A highly expressive way that respects the contextual information of the required knowledge is necessary. This could be provided by a knowledge object ontology describing the specific concepts to be used during knowledge retrieval. Furthermore a strong codification of human resources such as experts is necessary for the retrieval of knowledgeable people who could provide advice on the problem. In our example, the manager would ask from the service a document of the type “roadmap” or “best practice” that could be used to set up a new business in the country “China” and the “Automobile” industry. All the parameters in italics should be part of the knowledge represented in the knowledge object ontology and could be used by the knowledge provision service to retrieve the proper knowledge object(s). The same parameters could be used to infer which expert (e.g., a specialized business analyst or a lawyer with knowledge of Chinese business law) could provide useful advice.

From the above it is obvious that the main difference between knowledge provision services (which leverage representations of knowledge objects to achieve their efficient retrieval and composition and support effective problem solving and efficient decision support) and semantic Web services is that the former have many more descriptive features whereas the common thing between them is that they can be conceptualized using the same four types of features: access, descriptive, functional, and structural.8

In the present article, we provide a declarative specification of the knowledge representation schema used for describing knowledge objects and the related background knowledge. This is the basis for more context-specific characterizations of knowledge objects that will allow better search and retrieval for powerful new services, as well as easier composition within an encapsulated knowledge provision service. Concerning the conceptualization of knowledge provision services, we guarantee access features by the use of SOAP, description by using our knowledge object ontology, functioning by retrieving or composing the knowledge objects, and structuring by combining different types of knowledge objects.

The structure of the article is as follows. In the following section we present an ontology that we adapt for the description of knowledge objects, which we further use to enrich the descriptive features of the knowledge provision services. In Section 3, we present knowledge representation in knowledge provision services and its implementation as a modification of OWL-S. The architecture for delivering knowledge provision services and the respective mechanisms are analyzed in Section 4. The knowledge services registry and the composition service are presented along with a case study in Section 5. In Section 6, we discuss the implementation and evaluation of our approach. Finally, in Section 7 we discuss related work, and in Section 8 we present future work and concluding remarks.

2. THE KNOWLEDGE OBJECT ONTOLOGY

An ontology that describes knowledge objects will be used as the basis for the descriptive features of our knowledge provision service. This ontology was
introduced in our previous work, and we adopt it for the needs of knowledge object representation for knowledge provision in interorganizational systems. Such an ontology comprises:

- a specification of all attributes of a knowledge object for tracking down knowledge in an open environment
- the value ranges, and—if necessary—supplementing related ontologies, for defining the ranges of attributes used
- a specification of all links and relationships that may exist between knowledge objects (indicating, e.g., that some knowledge object could provide prior knowledge useful for understanding and applying some other knowledge object)

The knowledge object ontology (KOO) includes two basic facets: the content facet, which characterizes the content of the knowledge object, and the context facet, which describes the context in which the knowledge object can be applied. The idea behind this description is that, if the content and the context of a knowledge object are sufficiently described, it should be possible to assess the content and its potential usage comprehensively and consequently its real value for a user facing a specific problem.

In the following, we present a short overview of the content and context facets of the knowledge object ontology.

### 2.1. The Content Facet

The content facet is the core of a knowledge object. It contains the content of an knowledge object—if electronically available—plus metadata describing the kind of content, what it is about, and how it is physically manifested; see Figure 1. Attributes and values for these descriptions are defined by corresponding content ontologies.

As default attributes we provided the following ones that are directly taken from the Dublin Core (DC) or the Learning Object Metadata (LOM) standard, respectively, such that we do not explain them further in this article:

![Figure 1. The content facet of the knowledge object ontology.](image-url)
Now, in addition to these generic structures, there may be application-specific extensions and adaptations depending on the case. For example, the Content_Type_Object could be extended to include Case Studies, Methodological Tools, and so forth.

An extension of particular interest is the one that is made for expert knowledge. This kind of knowledge is tacit and cannot be easily codified. Yet, the expertise of a person could be represented as a knowledge object in the form of an expert profile. The profile is a Content_Type_Object that has its own content representation as well as contextual characteristics that comply with the following context facet description.

### 2.2. The Context Facet

The context facet describes the application context in which a particular information object can be used. First, we have to note that content and context may be different even if they are not necessarily both existing and different in concrete application examples. For example, in the case of consulting projects or lessons learned from consulting projects, it might be difficult to make a distinction between content of a project and potential application context. Second, we note that we consider the fact of having both content and application context at our disposal. The reason for this is that often a user does not know what the content of a knowledge object useful for him could look like. He does not know the answer, but he knows the problem. This means that what we need is describing problem situations (i.e., potential application context) and linking them to possible solutions.

Figure 2 shows the general structure of the description of potential usage context. It is composed from two parts:

- the static context that describes the organizational context in which a knowledge product may be applied
- the dynamic part that describes the concrete dynamic situation in which a knowledge product may add value
Figure 2. The context facet of the knowledge object ontology.
The organizational context shall describe as comprehensively as possible an intended consumer of a given knowledge object. Currently, we foresee the following attributes to realize such a comprehensive description:

- the intended User_Organization, if some knowledge object is produced exclusively for specific customers or its applicability depends on certain customer company characteristics, like the size, the location, or the legal form of a company
- the intended User_Department within this organization, because a given knowledge object may only make sense to be used by the marketing department or the production planning
- the Organizational_Role(s) that may apply a knowledge object successfully (because they have the competencies, rights, or responsibilities to do so or because a knowledge object—like a lesson learned or a best practice—affects in particular their specific job)
- the Age and professional Experience of the people in these Organization_Roles, because there might be preconditions that must hold to employ a knowledge object effectively (e.g., such conditions frequently exist in engineering environments)

On the other hand, the dynamic, situational context is constituted by following attributes trying to describe as detailed as possible what activity shall be executed in which manner. For this description, we oriented ourselves on the classical “W-Questions,” who, what, when, and so forth. We shortly summarize these attributes:

We describe which process (e.g., a certain production process) is performed, manipulating what entities (as input, output, or auxiliary products), under which conditions (e.g., obeying specific regulations with respect to health or environment), and to which purpose, by which people, through which means, and in which general application context (e.g., the industry sector). These dimensions can be further decomposed and instantiated/adapted for specific cases.

The next step is to incorporate the KOO into the knowledge provision services’ descriptive features.

### 3. OWL-S REPRESENTATION FOR KNOWLEDGE PROVISION SERVICES

There have been a number of efforts to add semantics to the discovery process of Web services. An upper ontology for services has already been presented to the DAML-S Project. According to it, the root of the ontology is the Service class and, underneath, the ontology is structured so as to provide three essential types of knowledge about the service:

- What does the service require of the user(s) or other agents and provide for them? The answer to this question is given in the service profile.
- How does it work? The answer to this question is given in the service model.
- How is it used? The answer to this question is given in the service grounding.

In knowledge provision services, the service profile provides a clear description of the knowledge object to be used, the service model involves retrieving the right knowledge object, and the service grounding the way the object is exchanged; see Table I.
As previously mentioned, knowledge provision services leverage organizational knowledge that is codified in knowledge objects. The representation of a knowledge object, and subsequently a knowledge service, is succeeded by the knowledge object ontology. Our knowledge provision service ontology is using the upper level nodes of the OWL-S ontology. The ontology incorporates all OWL-S concepts and attributes of the SERVICE GROUNDING. However, in terms of SERVICE PROFILE, it uses the OWL-S specification only as a skeleton to define the service, but the concepts that determine the service features are based on the knowledge object ontology. The SERVICE MODEL concepts are altered only in the sense that we apply retrieval methods.

The retrieval of a knowledge object is shown in Figure 3. The method retrieve will get as input at least one of the attributes of the context facet, let us say, as an attribute of the service profile.

<table>
<thead>
<tr>
<th>Semantic Web services</th>
<th>Knowledge provision services</th>
<th>Web services (WSDL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service profile</td>
<td>Knowledge object ontology describing the knowledge object to be retrieved by the service</td>
<td>Types and messages</td>
</tr>
<tr>
<td>Service model</td>
<td>We have implemented the “retrieve” operation</td>
<td>Operations</td>
</tr>
<tr>
<td>Service grounding</td>
<td>SOAP</td>
<td>Ports and bindings</td>
</tr>
</tbody>
</table>

Figure 3. The knowledge provision service ontology.
example, the industry. It will fetch all objects whose metadata match to the given input, namely, the concrete industry.

Yet, to make the retrieval more efficient, we also apply similarity measures. Semantic similarities in given contexts have been thoroughly studied both in database object level and also in ontology level.

In Ref. 10 Kashyap and Sheth use the concept of semantic proximity to associate domains under a specific context, whereas in Ref. 11 it is claimed that when trying to measure the similarity between two ontologies we have to pursue a two-layer view: the lexical layer and the conceptual layer. In the same work, Levenshtein Edit Distance, which is a well-established method for measuring the distance between two strings, is used for measuring lexical distance and similarity, whereas in the conceptual layer, similarities should be accounted in conceptual structures, namely, taxonomic and nontaxonomic relations.

In our approach, the KOO clearly determines taxonomies and taxonomic relations in the modeled domain. Therefore, similarities are used to provide better ranking between retrieved knowledge objects and not for ontology mapping or ontology searching. In our implementation, because ontologies have to comply with the same conceptual reference model, namely, the KOO, we choose to apply a simple but indicative method to compute similarities in the same taxonomies.

In the industry example, if the specific input has \( n \) subclasses in total and the returned knowledge object has in its annotation \( m \) subclass instances, then the similarity of the given object for the concrete parameter (parameter similarity, \( s \)) is set to \( s = 100 \times m/n \). If the retrieve method accepts \( r \) input parameters, then the total similarity will occur as the product of \( r \) parameter similarities, namely \( s = \prod s_i \), where \( i = 1 \ldots r \).

In the case that multiple ontologies are included, the previous similarity computation would prove to be rather naïve and less effective. In this case, multiple methods for computing similarities and ontology mapping could be considered. In Ref. 12, manifold examples and approaches are presented for multiple ontology systems.

4. ARCHITECTURE FOR DELIVERING KNOWLEDGE PROVISION SERVICES

Web services are currently described using WSDL descriptions. A WSDL file defines services using six major elements, types, message, port TYPE, binding, port, and service. OWL-S adds semantics to WSDL files by extending and relating existing WSDL constructs to OWL ontologies. In our approach, we implement a similar technique to achieve relations of ontological concepts to a WSDL file.

The main idea of our architectural approach is the use of the Onto2Wsdl Translator software, which we have developed as a Java software package. The Onto2Wsdl tool receives as input the KOO, which is used for annotating knowledge objects. Knowledge objects are stored in the knowledge repositories of firms interested in sharing and/or trading their knowledge objects. The output of the tool is a WSDL file describing a knowledge provision Web service of these firms’
repositories; see Figure 4. Currently the tool is running on an Oracle 10g application server, but it can be easily customized to run on any Web services-enabled application server.

The Onto2Wdsl is a three-tier application. On the integration layer we have the OWL-S knowledge object ontology previously described. On the business logic layer, we parse all concept and relation classes to Java classes through the use of a Jena parser. These classes include methods that execute queries to knowledge repositories through jdbc, validate queries against the ontology, rank results, and construct WSDL files on the fly.

A WSDL file that describes a knowledge provision service consists of types and messages that are mapped to the knowledge object ontology, a retrieval operation, and ports and binding, as defined by SOAP. For example, in the WSDL file in Figure 5, the part name businessprocess is mapped to the “which_process_is_performed” relation of the KOO through the Onto2Wdsl tool.

On the interface layer, the final WSDL files allow soap manipulation by Web services. For example, the above-mentioned WSDL file describes a Web service requiring as input the content type of the requested knowledge object and the business process that it is applied to. When a message with this input is received, the tool validates the input parameters against the ontology, queries the knowledge repository, and it finally provides as output the most relevant knowledge object.

5. KNOWLEDGE SERVICES REGISTRIES

With the help of the Onto2Wdsl tool, we have a means of publishing knowledge repositories through Web services. In the case of more than one knowledge repositories, we need a means of discovering the proper knowledge provision service to invoke. In the traditional Web service environment, this role is played by UDDI. UDDI provides a standards-based set of specifications for service description and discovery, as well as a set of Internet-based implementations.

Information provided by a UDDI registry can be conceptually divided into four categories, each one representing a top-level entity in UDDI: tmodels, businesses, business services, and service bindings. Every entity is assigned its own identifier and can always be located by this identifier in the context of the specific UDDI registry.

_tmodels can, under certain circumstances, support the addition of semantics in the description of a web service, as, for example, in Refs. 7 and 13. Nevertheless,
this support is limited for covering the expressivity needs of knowledge provision services. Hence, a more intelligent way of registering knowledge provision services is required. In our implementation, UDDI is replaced by a knowledge provision service registry (henceforth called KPSR) that supports richer and more appropriate semantics; see Figure 6.

Technically, publishing a knowledge provision service to KPSR is realized by the submission of its metadata to the registry’s database. Knowledge provision services’ metadata apply to the same faceted analysis as the one applied to knowledge objects (content metadata and context metadata). The metadata at this level, the service level, describe the available knowledge in the relevant repository. For example, a management consulting firm may provide case studies and diagnostic tools specialized for the automobile industry. The knowledge provision service

Figure 5. Extract of WSDL file as processed by the Onto2Wsd1 tool.

Figure 6. The Knowledge Provision Services Registry.
that publishes the consultancy’s repository should describe the specific content of the repository, namely, it should describe itself or else what it delivers, by using the appropriate metadata. These metadata comply with the upper level of the KOO that extend the SERVICEPROFILE. The service’s other features, namely the SERVICEMODEL and the SERVICEGROUNDING, remain intact, as shown in Figure 3. In case the repository is changed, for instance, being enriched by methodologies or covering another industry, the registry’s database is updated, respectively, with the appropriate metadata.

The discovery of knowledge provision services is provided by the respective discovery service of the registry, which is exposed via a Web service interface. Queries to the registry are ontology enabled and are made feasible by using the same tool (Onto2Wsdl), as when querying repositories for knowledge objects. Table II presents the main differences between a UDDI and a knowledge provision services registry.

Apart from the discovery service, a KPSR also provides a composition service; see Figure 7. In the context of knowledge provision, composition is understood as the result of the aggregation of several knowledge objects retrieved from different knowledge repositories. We refer to this aggregation as a composed object. This process is analogous to the bundling of news items customized to the user’s preferences. A composed object may consist of knowledge objects of various, different or the same, content types and from different providers. For example, in the case of the new business in China scenario in automobile industry, the composition of several objects from various knowledge provision services could be required for fulfilling the manager’s specific needs. The knowledge objects to be composed in this case could be a road map to investments in China, which could be provided by the Chinese Ministry of Development, a best practices guide, provided by a consulting company that may have facilitated similar projects in the past, a handbook specifying all the actions for acquiring a license to create a new business in the specific country, provided by the local Chamber of Commerce and probably a lawyer to provide the legal advice that cannot be found in books.

The composition service receives as input the user’s request that is provided in the form of a template. The user fills in the template with the description of the types of content she or he searches for, for example, road maps and handbooks, and the context of application, for example, setting up a new business in China. Then the template is transformed into a WSDL file that describes the request. Then it is split up in as many WSDL files as types of content are specified in the template. In the case that the user leaves content types undefined, the same procedure

<table>
<thead>
<tr>
<th>Table II. Comparison of UDDI and KPSR.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UDDI</td>
<td>KPS Registry</td>
</tr>
<tr>
<td>tModels</td>
<td>Metadata references</td>
</tr>
<tr>
<td>UDDI identifier</td>
<td>Unique key to each knowledge service</td>
</tr>
<tr>
<td>Does not supports semantics</td>
<td>Enhanced by OWL-S and KOO</td>
</tr>
<tr>
<td>Does not supports composition</td>
<td>Supports knowledge-based composition</td>
</tr>
</tbody>
</table>

International Journal of Intelligent Systems DOI 10.1002/int
is repeated for all content types recorded in the registry’s database. In the previous example, from the initial WSDL file four WSDL files will emerge, one for the road map content type, one for the best practice guide, one for the handbook, and one for the legal advice. Respectively, four queries are executed and the registry’s database is searched for the most appropriate knowledge provision services. When the services are found, they are invoked and relevant knowledge objects are retrieved.

The next step concerns bundling the most appropriate knowledge objects into a composed one that will support users solving their problem. There are two options here. First, the user can view the retrieved knowledge objects, which are presented per content type and are ranked according to similarity, and compose manually a bundle that fits her or his needs by selecting the most useful knowledge objects according to her or his valuation models.

Second, the composition service may propose a set of composed objects, which are automatically created on the basis of specific criteria. The criteria we have included at the moment are reduced just to similarity. Similarity has been discussed in the section presenting the retrieval service. According to this criterion, during the retrieval process the knowledge objects that fail to pass a threshold are discarded. The composition service generates an initial set of composed objects by bundling the selected knowledge objects and additionally assigns to the composed objects a similarity measure, which is calculated by simply multiplying the similarities of the componential knowledge objects. Respectively, composed objects that fail to pass a threshold are discarded. The user may select any of the ranked composed objects and enhance them with knowledge objects that have not been included by the composition service, yet have been retrieved by the selected knowledge provision services.
Knowledge composition in distributed environments is a challenging research area with several unsolved problems. Here we presented the basic infrastructure and a simple composition technique that we aim to enhance in the future, as we discuss in the final section.

6. IMPLEMENTATION AND EVALUATION OF OUR APPROACH

To provide a rich and semantically powerful mechanism for describing knowledge intensive services, we have created knowledge models to enable the discovery and delivery of appropriate pieces of knowledge in a specific context. The resultant knowledge models that adopt the KO ontology have undergone a 6-month test in a management consulting firm, in the course of the INKASS IST research project. Throughout the test, the users perceived the ontology as being very helpful in classifying their questions (average rating 4.3 in a Likert scale from 1 to 5, with 5 representing the maximum satisfaction), and the quality of the problem solutions provided by the system was perceived as high in terms of information quality measured in relevancy of the documents provided in the context of the question asked.

Overall, the KOO knowledge representation has proven to add value in the business processes of the firm by organizing the knowledge inside the firm and defining a space of specific consulting problems. The combinatorial approach, grounded on the content–context breakdown of client queries, resulted in hierarchies (ranking) of potential knowledge for answering specific client requests or to composed objects that could constitute sounder solutions to client problems. In particular, in the consulting domain, the content and context of a problem (e.g., the involving business process or industry, see Figure 8) can be codified, located, and used by our infrastructure for specifying or structuring knowledge objects that could be used for providing solutions.

Figure 8. The results of a user query organized by the type of process it relates to. The taxonomy on the left frame includes the various types (industry, content, etc.) that can be used to reorganize the results.
The structure of knowledge objects in the latter case occurs as a composition of different, complementary content types. Nevertheless, it has advantages compared to a ranked list of different content types, as it offers a unified knowledge representation mechanism in the form of a single object called composed object. In the case of a composed object, advanced handling and managing operations, for example, pricing and negotiation, can be implemented easily using the proposed OWL-S-based representation formalism. In other words, an infrastructure is available and provides all the necessary knowledge representation aspects to enable the design and operation of various and complex service models, such as pricing of composed objects or nested objects where one calls and implements the other (in the consulting domain, a consulting methodology may be implemented by a software tool and may require an existing case study for comprehension reasons). These service models can add value to the composed object that goes far beyond a simple aggregation of content types.

During the implementation of our approach, we have collected and annotated various content types that could be used by consulting firms. These content types included consulting methodologies, case studies, manuals and links to software tools, CVs and availability of experts, databases, and folders and populated a pilot knowledge repository. The use of the Jena parser and the OWL representation of KO ontology (see Section 4) have allowed the implementation of our knowledge models as instances of ontology standards of Web services. In this way, we have achieved interoperability in a semantic level by using simple SOAP messages, hence building real-life knowledge provision with intelligent e-services.

Consequently, the knowledge service registry can host more intelligent and ready-to-use services; thus it broadens the types of Web services that are hosted by a traditional UDDI registry. In such a registry, other knowledge provision firms (e.g., consulting firms or freelance consultants) could be attached by simply complying with our formalisms and by using any infrastructure respecting common interoperability standards as SOAP.

7. DISCUSSION AND RELATED WORK

There exist several attempts to provide semantically enriched services. OWL-S (former DAML-S)\(^7\) employs semantic Web technology for service description. Competitive approaches to OWL-S are the Web Service Modeling Framework (WSMF),\(^14\) which aims to provide a comprehensive framework to achieve automatic Web service discovery, selection, mediation, and composition into complex services, that is, to make semantic Web services a reality and to exploit their capabilities, and the Meteor-S Web Service Annotation Framework,\(^15\) which attempts to add semantics to the complete Web process life cycle by providing constructs for adding semantics to current industry standards.

Although OWL-S is a powerful way to describe common semantic Web services, like a flight-booking or book-selling service, when it comes to knowledge provision services, it lacks modeling primitives that are deemed important for describing a knowledge provision service. Actually, we claim that OWL-S can

\(\text{International Journal of Intelligent Systems} \quad \text{DOI} 10.1002/int\)
provide the technological means to implement knowledge provision services under
the condition that it incorporates ontological primitives that adequately represent
knowledge objects. Parts of these primitives have already been conceptualized in
previous work.9

Table III presents the attributes describing the service profile of an OWL-S
service. Knowledge services involve the implementation of knowledge assets in
real-world services. The serviceParameter and serviceCategory attributes of the
OWL-S serviceProfile may be considered as the only contextual characteristics
of the service, apparently poorer and inadequately descriptive for the description
of knowledge objects. The limited parameters of OWL-S service Profile are, in our
approach, enriched and expressed in a clear and standardized format. We attempt
to initiate the dialogue for setting up standards that will provide expressive and
powerful semantics for dealing with knowledge intensive objects and services.

All previous approaches provide concrete models for the description of semantic
Web services that impose the analogous adaptation of traditional service pro-
cesses, like discovery and composition. Toward this direction, better and more
efficient service discovery has been achieved by matchmaking using specific algo-
rithms,13 faceted taxonomies,17 and ontologies.18 Furthermore, composition services
have been developed extending discovery mechanisms using the service-oriented
approach, like Refs. 8 and 19–22 or using P2P collective networks.23

The common ground between the above-mentioned approaches is the provi-
sion of more efficient searches for services focusing on interoperability and inter-
connection, toward which they employ WSDL7,15 or WSDL-like protocols.

The differentiating point of our work is that we primarily aim to provide a
well-formed and robust description of a knowledge provision service by defining
more accurate attributes of the service rather than improving the searching mech-
anisms on existing attributes. As a result, the declarative part of the service should
be (and is) much more detailed, introducing new mechanisms for service recogni-
tion, discovery, and composition. By this approach, we aim to facilitate knowl-
edge organizations in implementing their knowledge sharing and dissemination
strategies on the basis of a common technology and interoperability standards.
Our vision of a knowledge provision service is to act as an operational module in
an intelligent interconnection environment, for example, a knowledge grid, that
would enable individuals to manage knowledge resources easily and efficiently in
an automated way.

<table>
<thead>
<tr>
<th>attributeName</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serviceName</td>
<td>The name of the service</td>
</tr>
<tr>
<td>textDescription</td>
<td>A human-readable description of the service</td>
</tr>
<tr>
<td>contactInformation</td>
<td>No range description</td>
</tr>
<tr>
<td>serviceParameter</td>
<td>Allows pointing to a parameter and specifying also the value of the parameter</td>
</tr>
<tr>
<td>serviceCategory</td>
<td>Specify the categories to which the service pertains</td>
</tr>
<tr>
<td>OWL ontologies</td>
<td>May be added according to the case</td>
</tr>
</tbody>
</table>

Table III. OWL-S service profile attributes.16

International Journal of Intelligent Systems  DOI 10.1002/int
8. CONCLUSIONS AND FUTURE WORK

In this article, we have presented an approach for publishing knowledge repositories using semantic Web services as a vehicle. We have extended OWL-S with KOO, which represents knowledge objects in a generic, application-neutral way, to enable the expressivity that knowledge provision services require for their discovery and composition as well as the efficient knowledge retrieval. The contribution of our work is manifold:

- It provides an open interface to knowledge repositories, offering a gateway to the knowledge inside an organization firm, independent of its type, explicit or tacit.
- It introduces an ontology that provides the means for efficient retrieval of knowledge from distributed, heterogeneous resources.
- It develops the infrastructure for a knowledge provision services registry, which offers mechanisms for publishing, discovering, and composing knowledge provision services that will provide users with relatively complete knowledge for solving their problems.
- It develops a tool that translates ontologies in OWL format to WSDL files and vice versa without having to define necessarily a new language.

We are currently working on developing proper graphical interfaces for enabling users to annotate and register knowledge objects and services. Regarding knowledge composition, more criteria will be introduced, for example, the complementarity of knowledge objects for the formation of a solution to a problem, which complementarity can be assessed on the basis of the KOO.

Furthermore, in our future plans, we intend to treat the whole life cycle of trading knowledge in open environments. At the moment, knowledge provision services support mainly retrieve and compose operations. Thus, only knowledge sharing is possible. To support the full trading life cycle of knowledge objects, we aim to develop operations for pricing, contracting, and monitoring knowledge transactions. A similar approach for information goods is presented in Ref. 24. Our goal is to develop a concrete language for knowledge trading. Finally, another challenging direction for this work is the transformation of the computational model of the KPS Registry to a P2P infrastructure. A similar approach is presented in Ref. 25, where DAML-S is used for P2P discovery.

Acknowledgments

This research was partially supported by the European Commission under contract IST-2001–33373. The authors thank the anonymous reviewers for their detailed, accurate, and helpful comments.

References