Improving e-Services Adoption With Semantic e-Services Platform and Question-Based Approach

Luka Pavlič, Marjan Heričko, Maja Pušnik
Faculty of Electrical Engineering and Computer Science
University of Maribor
Smetanova ulica 17, 2000 Maribor, Slovenia
{luka.pavlic, marjan.hericko, maja.pusnik}@uni-mb.si

Abstract. *E-services have significantly changed the way of doing business in recent years. However, the implementation of these services remains poor. There is a significant gap between supply and actual e-service usage. This is why we started a project to provide an environment that will encourage the use of e-services. We believe that merely providing e-services does not automatically translate into consumers using them. This paper shows the origins of our project and its current position. We discuss the decision of using semantic web technologies and their potential to improve e-services usage. In this paper, we will present an e-services platform. Ontologies and semantic web technologies are used heavily in the platform. They enable the platform to be used as a basis for intelligent components, such as an e-service proposing component.*

Keywords. E-Services, E-Services Repository, Ontologies, Semantic Web, Guided dialoque

1 Introduction

The e-services (IT-supported services, e-services are typically available via a computer network) market has grown considerably. In the case of the Republic of Slovenia, this is also clearly stated in European Commission report for digital economy i2010 [1]. Although the report is limited to e-government services, it also provides good insight into the whole area of e-services. The e-service selection is satisfactory for both businesses consumers and individuals. However, we must note a low level of use of these services: e-services are used in less than three quarters of businesses consumers and only little over quarter of individual users [1].

Obviously, there is a large gap between supply and actual e-service usage. This is why we started the project "Ontology-based E-Services Adoption Improvement". Its aim is to provide an environment that will encourage the use of e-services. One of the most challenging features that the knowledge management area is trying to address is the search for knowledge. The most popular searching approach includes browsing categorized knowledge assets (with the support of taxonomies) and full text searching [2]. We believe that today's modern technologies and knowledge management approaches have the capabilities and opportunities to support more intelligent knowledge searching. In this paper, however, we will not address knowledge searching in general. We will focus on searching e-services in a well-defined situation. We established a platform, where searching becomes proposing. With the platform we enabled technology, where a problem solution can be proposed by the "searching" engine. The platform enables the search for a knowledge asset with the possibility of having an explanation of the solution. From the user's point of view it is used regardless of the knowledge representation structure or the location in the network.

This platform is potentially useful in reducing the gap between the supply and the use of e-services - we are proposing an appropriate method of formal presentation of knowledge about e-services and applications on this basis. We believe that formal notation is necessary to significantly improve the use of e-services. At the same time, we will ensure that the new notation will not demand any additional activities by e-services users or providers. To enable our methodology, we have built a prototype platform. It allows users to use advanced components, such as an intelligent proposing component to discover an appropriate e-service or combination of them. The system will also allow for verification, and to check if there is any alternative e-service or a collection of e-services to be selected.

This paper presents the current state of the semantic web-enabled platform, its core structure and the potential for improving e-services adoption. In the section 2, we present the most important related attempts to cope with presented problem. In section 3 we explain why having formal service descriptions using ontologies is so important. We also present our ontology and its meaning. An application of this representation is shown in section 4, where platform and functionalities are described. As example we also show an intelligent component, which can use expert knowledge in out ontology to lead dialogues with users. The platform is a work done at the moment of writing, however a lot of work still has to be done, especially in terms of validations. This is described in section 5. We conclude this paper in section 6 with summing up key contributions and their potential to improve e-services adoption.

2 Related work

The gap between the high amount of offers and the low level of e-service usage has been noted and addressed by many authors. The problem is not local, since it has been observed throughout the European Union and in other IT developed countries.
Wang et al. [4] proposed the use of a specific vocabulary, based on ontologies, which would allow an intelligent search and automatic integration of services on the Internet. The author is limited only to online services in the technical sense (“pure” Web services with software interface and without a user interface).

Certain authors have already partially addressed the search of e-services. Shan-Liang et al. [5] discovered that just indexing keywords cannot serve as a search for appropriate e-services. They therefore proposed the statistical ontological matching of e-services. This could have been the basis of intelligent e-services inquiries. The study is interesting because the authors are not limited to Web services, but to e-services as a whole. However, they request a specific e-Service interface (OWL-S).

There are also more attempts at using ontologies as a helping tool. For example, article [6], which proposed the ontological description of Web services. For each domain it suggests its own ontology, which provides a basis for the Web agent’s search for appropriate e-services. Thus, accumulated knowledge can be used to facilitate the search for e-services. The author only addresses the collection of data about services in certain domains, and does not address the searching of e-services.

3 The role of ontologies and semantic web technologies in the platform

The idea of the semantic web allows for automatic, intelligent inferring of knowledge, supported by ontologies. The basic idea of the semantic web is a different organization and storage of data and, subsequently, new possibilities for using this data [3]. With the semantic web, data is not presented as fixed and eventually connected data assets, such as in relational databases. It enables the connecting of knowledge assets of many types in a very free way. The barrier that prevents the more advanced usage of available data is believed to be the semantic poorness of today’s solutions. The result is an inability to make real use of the enormous amount of available “knowledge”. The semantic web gives us the opportunity to overcome these difficulties. The concept of meta-data was introduced into the core of the semantic web. Using meta-data, so called smart agents can be used to search for information by content and to make inferences based on the gathered concepts. As a foundation, there has been a lot of work done with regard to common formats for the interchange of data and the common understanding of common concepts. This not only allows a machine to perform some intelligent tasks on data automatically, but also for a person to browse, understand and use data in a more straightforward way. Furthermore, semantic web ideas can be used in an internal enterprise information system for knowledge management in a different way to introduce new intelligent services. Knowledge, presented using the semantic web technique is distributed by default – there is simply no difference if all knowledge sources are local or remote. For the stack of semantic web technologies, please see the Semantic Web Initiative web pages [8].

One of the enabling approaches used in the semantic web is metadata. It is supported by the concept of ontologies and has its foundation in W3C standards. Ontology describes the subject domain using the notions of concepts, instances, attributes, relations and axioms [7]. Among others, concepts can also be organized into taxonomies, whereby inheritance mechanisms can be used in an ontology. Ontologies are built on description languages, such as RDF(S) and OWL, and add semantics to the model representation. Their formal, explicit and shared nature makes them an ideal foundation for presenting knowledge in our platform.

The platform’s expert knowledge (i.e. question-answer pairs) is expected to be expressed according to prepared ontology (see Figure 1). With some of the presented facts, we can also justify our decision to use ontologies as well as other semantic web technologies to provide a basis, not only for expert knowledge descriptions, but also for future intelligent services:

- Ontologies in the semantic web have their foundation in W3C standards.
- Ontology-based knowledge descriptions are computer readable and therefore suitable for automated (computer) processing.
- Transforming OWL and RDF based knowledge representation into other forms of representation (in textual or graphical form) can be achieved easily with relatively simple transformations.
- Enabling technologies are well established, recognized and extendable.
- They enable knowledge exchangeability in a straightforward way.
- The semantic web introduces technology that enables knowledge to be distributed.

As described, ontologies allow us to focus on the user and enable knowledge to be used automatically by an intelligent proposing component. Knowledge that is required by the proposing component is expected to be consistent with the ontology, as presented in the platform (see Figure 1).

A core platform's ontology fragment is shown in Figure 1. We introduce the concept of the Solution (Service in the case of e-Services platform). Since we include the possibility of related solutions at a platform level, proposing a component is able to propose not only potentially interesting solutions, but also solutions that can be related to the proposed one.

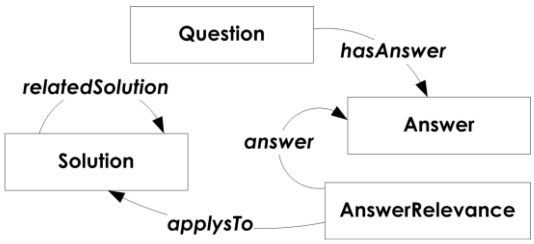

Figure 1. Ontology structure, expected by the platform.

The core of our e-services-enabled ontology is shown in Figure 2. We have classified services to be simple services, not supported by ICT (Information Communications Tecnology) or e-services. We also manage services, which are composed with several services. The ontology also covers special e-services, supported by web applications or exposed with web services. An ontology enables the capturing of several service providers. They can also be classified using taxonomies and folksonomies. Please note that Figure 2 only provides a basic insight on some classes and their relations within a larger ontology.

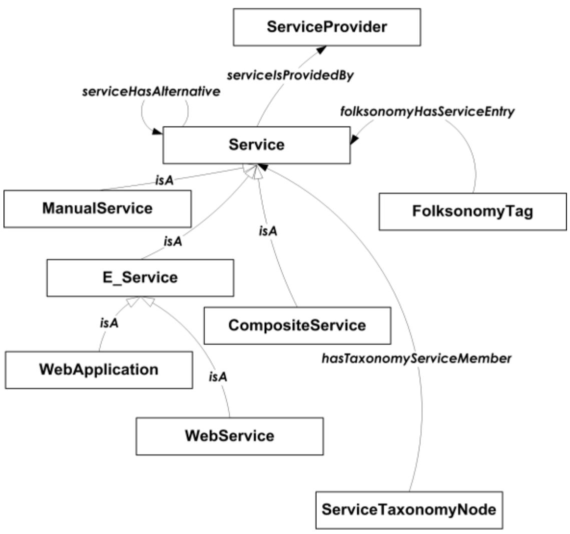

Figure 2. Core of defined ontology.

In the platform, we have also used a novel approach for working with ontologies. All OWL, RDF and SPARQL-related operations are achieved using the Jena framework. In addition, we also developed our own solution for mapping ontologies to classes and objects in Java. For this purpose, we developed the basic components that allow basic work (CRUD – Create, Read, Update and Delete operations) with distributed data on the one hand, and their ease of use across the platform on the other. It is used in a similar manner as the JPA library (Java Persistence API) - with tags classes and methods. The core class diagram is shown in Figure 3.

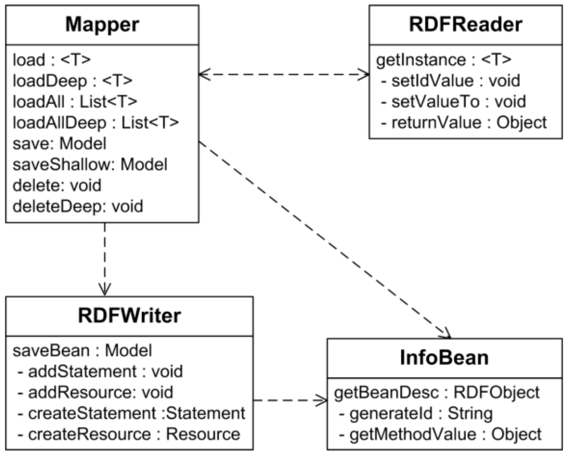

Figure 3. Object-Ontology mapping class diagram.

The "Mapper" Class enables all mapping functionalities. It allows data to originate from any location on the web, databases, files, etc. The component allows you to create classes based on ontologies and vice versa. In particular, it allows the mapping of RDF instances of objects and vice versa. For this purpose, we have developed appropriate tags (@OwlClass, @Id, etc.).

4 Platform functionalities and design

4.1 General purpose functionalities

The prototype implementation of the platform is based on the formal description based on ontologies. The prototype enables the hosting of intelligent services, which will empower the use of e-services. The platform will also be available through application programming interfaces (API). In this way, the prototype itself will be an e-service available for reuse in third party applications. The prototype is implemented using open-source and freely available software. The base of the platform is semantic web technologies.

The platform itself, among other things, allows knowledge integration from distributed data sources (catalogues of e-services, World Wide Web, etc.) and the management of the gathered knowledge. Additionally, the platform also:

- enables providers and users of e-services the annotation of services with additional knowledge,

- integrates data from the World Wide Web, and from additional data sources,
- transforms data encoded in the RDF language to a user-friendly format,
- manages an index of words, which were used in the platform and additional data sources in order to enable searching based on keywords,
- provides all data in raw RDF format, which will in turn enable newly built components to easily reuse information,
- manages a collection of real-world examples.

The prototype implementation contains 3 services, which runs on the presented platform (see Figure 4):

- full text search (data is not only indexed from local sources but also from external sources, e.g. World Wide Web),
- proposition of e-services or collection of e-services by using questions and answers approach,
- search for alternative e-services or alternative collections of services.

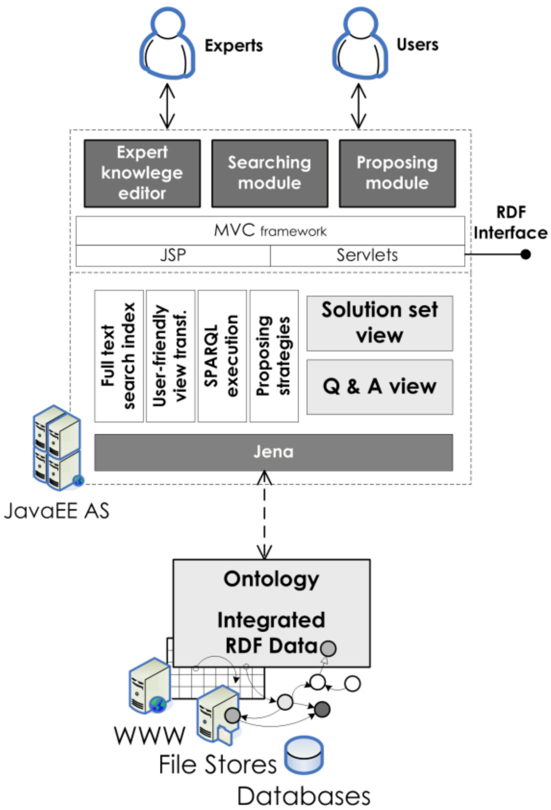

Figure 4. Platform architecture.

All knowledge, held or integrated by the platform, is also available through the RDF interface. This enables possible future integrations with similar systems in a relatively straightforward way.

4.2 Proposing intelligent component

The platform only expects some question-answer pairs, and then connects possible knowledge sources, the rest is handled by the platform itself in a so-called proposing component. It combines questions into a guided dialogue with users. The aim of the proposing component is to combine a sequence of questions used in a dialogue dynamically. We want to ask the user as few questions as possible to get enough information on the problem situation and consequently to propose possible solutions. Figure 5 captures a bird's eye view of our approach: domain experts contribute simple question-answers pairs, which are used in a guided dialogue with users during the solution advisement procedure. The solutions proposed by the proposing component are then presented to the user as potential ones.

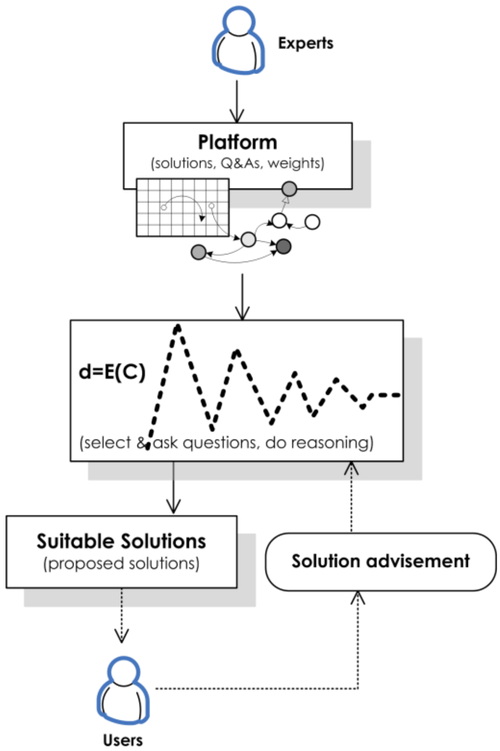

Figure 5. Bird's eye view of the platform for identifying suitable solutions.

The intelligent proposing component allows us to avoid the creation of static decision trees. In our opinion, ontologically encoded knowledge provides more potential for the integration of distributed data. The construction of decision trees, on the other hand, is a labor intensive task, which requires a high amount of input.

Proposing strategies enable the selecting of questions to be asked. The progress of the dialogue is measured constantly. Proposing strategies are currently based on two measures of success: entropy-based measures from information theory [9] and our own measures. During the dialogue between the platform and the user, a measure of success is used to verify what question should be asked or if another question needs to be asked at all. We are basically selecting questions that would improve the solution selection possibility in even the worse case. One of the first measures tested was a measure based on entropy theory. Later we introduced our own measures, since the entropy-based measures demonstrated in experiments that they produced far more questions in sequence than expected. Our own measure, which takes into account how many candidates were investigated and how many are possible to succeed as real candidates, was shown to generate less questions asked at the same success rate. It is not our intention to compare these two strategies in details here.

Figure 6 shows the core of the algorithm for guiding a dialog with user. It expects a set of e-services as a vector $P = [p_1, p_2, p_3, ..., p_n]$, where every service is presented with p_i $(1<=i<=n)$. The number of all possible services is n.

In the algorithm itself and as an output, the vector C is used. Its structure is as follows: $C=[c(p_1), c(p_2), c(p_3), ..., c(p_n)]$. Let us call vector C a “certainty vector”, where $c(p_i)$ is the current certainty for service p_i. The vector values $c(p_i)$ are within the range [0,1]. The number 0 means that it is certainly not likely that the service is the final solution, 1 means it is certainly likely that the service is the final solution, while with 0.5 we have no information about the e-service whatsoever. It is obvious that the initial value of vector C is $c(p_i)=0.5$ for all i. It is the goal of the algorithm to employ questions until vector C is transformed as much as possible.

When vector C is evaluated with a value that is deemed good enough (at the moment it is the experimental determined threshold), the algorithm is finished. If this is not the case, the next most promising question is selected.

When the answer is given, vector $A=[a(p_1), a(p_2), a(p_3), ..., a(p_n)]$ is constructed. The values $a(p_i)$ are the values of relevance for a particular service in the given answer. For e-services that are not mentioned by answer, we put value 0.5 (which means that we have no information on the e-services). When changing vector C (current certainty vector) with regard to vector A (the current answer), several guiding factors should be followed, which are captured by Equation 1.

We designed the algorithm (Figure 6) in such a way that the evaluation method is separated from the algorithm itself. As mentioned before, this is how we were also able to compare several competing formulas. Our final formula for calculating the relevancy of vector C is calculated as shown in equation 2.

$$c_{new}(p_i) = \frac{c(p_i) + a(p_i)}{2};\ a(p_i) \neq 0.5$$
$$c_{new}(p_i) = c(p_i);\ a(p_i) = 0.5 \qquad (1)$$

$$Rel(C) = 1 - (c_{max} - \bar{c}) \cdot D$$
$$D = \frac{4 \cdot (n-1) - 3 \cdot count(c_{05}) - count(c_{max})}{4 \cdot (n-1)} \qquad (2)$$

In equation 2, c_{max} is the maximum value of vector C, $\bar{c}$ is the average value of vector C and D is the factor that captures the current »progress« of vector C: $count(c_{05})$ means the number of e-service certainties, that share the value of 0.5 and $count(c_{max})$ means the number of e-service certainties, sharing the maximum value in vector C.

```
create a set of all questions QUE
create a set of all e-services P
guide_dialog(QUE, P)
  THRESHOLD=0.45
  create vector C,
    ci=0.5, 1<=i<=n, n=number of e-services
  while QUE not empty
    Q=select_question(QUE,C)
    if (Q==empty) print C, terminate
    remove Q from QUE
    collect answer ANS
    if (ANS == “do not know”) continue
    create vector V_ANS from answer ANS
    C=combine(C,V_ANS)
    print C
    if (user select to finish) terminate
    if (rel(C)< THRESHOLD) terminate
  do
end
```

Figure 6. The e-services proposing algorithm

The algorithm as a whole works like follows. The function “guide_dialog” is the primary one (see figure 6). It is used to select questions until termination by a) the user b) the solution is selected or c) there are no more questions. Since the user observes the progress of vector C all the time, it is easy to see when a particular e-service is starting to gain. The user is also given the opportunity to skip questions that he or she does not understand or does not know the answer. The threshold value of 0.45 for $Rel(C)$ was determined in experiments and was shown to work well. This value can, however, be easily changed in order to investigate its ideal value.

The function “select_question” (not presented in details) receives a list of all unasked questions and the current state of vector C. The output is a question to be asked, if possible. It is selected according to formulas described before: selected question would lower the value $Rel(C)$ (equation 2) even if the worse answer is given. This approach is also well known and proven in the game-playing theory.

There is also the possibility of implementing another strategy of asking questions. One could ask random questions until a measure of success meets a certain threshold. Another could also build a fixed decision tree based on question-answer pairs and decisions available. This strategy is actually available in the platform to support a comparison of a human-designed question sequence with calculated ones.

5 Further work

It is our goal to provide an e-services management methodology at the end of the project. In addition to e-services, repository management also allows advanced, easy-to-use mechanisms for the search and proposal of e-services. The project outcomes will allow user collaboration and linking of e-services. At the moment, the platform itself is almost complete, so we can already focus on researching its benefits.

Firstly, the methodology approaches and the platform will be demonstrated in real life. We will address e-services that are targeted at the student population.

The work on the project will be verified using established scientific research methods. In the final stages of the project, we will conduct a controlled experiment and a large number of surveys on the test population. In this manner, we will determine whether and, to what degree, our approach helps promote the use of e-services. At this time, our platform will go public.

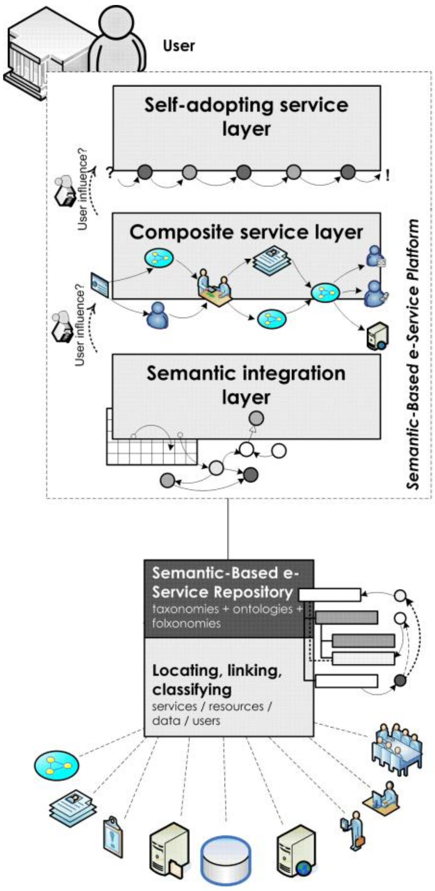

Figure 7. The final goal of the plaftorm.

The presented platform is actually part of a larger attempt to make e-services usage an easy task for everyone. Based on the functionalities that presented in this paper, we are planning to further enable even more advanced functionalities (see Figure 7). Those include:

- Easy e-service composition in a semi-automatic manner,
- Exchanging composed e-services between users,
- Using platform facilities to automatically select the most appropriate e-service in the case of (technical) failures.

6 Conclusions

In this paper, we have presented a gap between supply and actual e-service usage. This is why we started a project to provide an environment that will encourage the use of e-services. This paper showed the origins of our project and its current position.

We discussed the decision of using semantic web technologies and their potential to improve the use of e-services.

In this paper we also presented our own original approach to finding solutions. It is based on minimal expert involvement and the intense involvement of modern, intelligent technologies. Among other things, we also introduced a component for proposing the appropriate solution in a problem situation. The platform that we built for this purpose is based on ontologies and semantic web technologies and was built using open-source frameworks and libraries.

Based on the research of existing formal representations, and based on the research of ontologies and Semantic Web technologies, we have presented our own approach for describing knowledge about e-services. Our own knowledge representation technique can be seen as a necessary extension, which allows greater search capabilities and a higher grade of use for e–services.

At the end of the project we will provide an e-services management methodology, which will in addition to e-services repository management also allow advanced, easy to use mechanisms for the search and proposal of e-services to be used. The project outcomes will allow user collaboration and the linking of e-services. The platform itself is almost complete.

At the moment, the project offers some original contributions. The most important are:

- an ontology-based method for describing e-services and knowledge about them,
- a holistic methodology, which covers the capturing, management and using knowledge of e-services,
- a semantic web-based prototype platform for hosting intelligent components based on knowledge about e-services,
- collected knowledge about ready-to-use e-services.

At the moment of testing we can summarize that even at this stage, the outcomes of the project are promising.

References

[1] i2010 Report | EUROPA, http://ec.europa.eu/information_society/eeurope/i2010.

[2] D.E. Goldschmidt, "A comparison of keyword-based and semantics-based searching". Rensselaer Polytechnic Institute, 2005.

[3] T. Berners-Lee, J. Hendler, O. Lassila, "The Semantic Web", Scientific American, 284 (5). p. 28-37. 2001.

[4] X. Wang, J. Bernd Krämer, Z. Yi, A. Wolfgang, "Representation and Discovery of Intelligent EServices", Intelligence (SCI) 37, pp. 233-252, 2007.

[5] P. Shan-Liang, M. Qing-Jiao, Z.Ying-Xin, "A Hybrid Web Service Selection Approach Based on Singular Vector Decomposition", Services - I, 2009 World Conference, pp.724-731, 2009.

[6] J. Xiaogang, "Research on Web Service Discovery Based on Domain Ontology", Computer Science and Information Technology, pp.65-68, 2009.

[7] T.R. Gruber, "Towards Principles for the Design of Ontologies used for Knowledge Sharing", In N. Guarino & R. Poli (eds.), International Workshop on Formal Ontology, Padova, Italy, 1993.

[8] W3C, "Semantic Web", http://www.w3.org/2001/sw/.

[9] C. Shannon, W. Weaver, "The mathematical theory of communication", University of Illinois Press, 1949, USA.