



HAL
open science

Bringing Socialized Semantics into Web Services Based on User-centric Collaborative Tagging and Usage Experience

Uddam Chukmol, Benharkat Aïcha-Nabila, Amghar Youssef

► **To cite this version:**

Uddam Chukmol, Benharkat Aïcha-Nabila, Amghar Youssef. Bringing Socialized Semantics into Web Services Based on User-centric Collaborative Tagging and Usage Experience. The 12th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT-11), Oct 2011, Gwangju, South Korea. pp.450-455, 10.1109/APSCC.2011.19 . hal-01354543

HAL Id: hal-01354543

<https://hal.science/hal-01354543v1>

Submitted on 21 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Bringing socialized semantics into Web services based on user-centric collaborative tagging and usage experience

Uddam CHUKMOL, Aïcha-Nabila BENHARKAT, Youssef AMGHAR

Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205
F-69621, France

{uddam.chukmol; nabila.benharkat; youssef.amghar}@insa-lyon.fr

Abstract—The semantics of Web services is often provided by suppliers and embedded physically inside the service descriptions expressed in WSDL. This type of annotation can be very costly and is generally performed by well-trained cognitive or knowledge engineers. In this research work, we aim at providing an alternative to the actual Semantic Web Service (SWS) technologies by bringing a community-based or social semantics into Web services. This semantics is created by users through collaborative tagging on the capabilities of Web services after testing or using them. We argue that this type of semantic annotation can be very pragmatic and facilitate the process of Web service discovery, composition or clustering assisted by users.

Keywords- *Social Web service; user annotation; collaborative social semantics*

I. INTRODUCTION

Semantic Web community proposes numerous formalisms based on logics or ontologies to extend standard Web service description expressed in WSDL such as OWL-S [1], WSMO [2], WSDL-S [3], SAWSDL [4] or WSMO-Lite [5]. Semantic annotations based on these formalisms can be done at the service supplier sides only and may represent some cost and rigidity to them. In effect, modifying and maintaining ontologies are very complex and time-consuming tasks and in order to master completely the annotation process, the suppliers must acquire sufficient expertise in cognitive or knowledge engineering. In addition to the heterogeneity of semantics representations that may provoke several interoperability issues, the absence of fully semantic supported Web service registries in business environment discourages even more the enterprises in adopting SWS at large scale [6]. Besides, the SWS community itself reveals that there are still many challenges to convince practitioners to adopt their technologies [7][8]. At the user sides, exploiting SWS technologies means at least a proficiency in description logics or knowledge engineering, which can be considered as a burden. Moreover, there are not many tools supporting these technologies that are robust and in a form that developers find attractive and easy to learn and use [8].

The recent emergence of Internet of Service [9][10] and the increasing number of public accessible Web services in the Internet [11][12][13] through portals or catalogs lead us

to several other research issues. The main one that interests us is how to assist users at best in the discovery and composition of Web service in the aforementioned environment (publicly accessible Web services in the Internet). We are aware of the limit met in the keyword based service discovery systems due to the poor textual descriptions that can be extracted from WSDL descriptions of Web services [12]. We assume also that users who browse Web service portals in the Internet are not proficient enough in constructing discovery queries using description logics or SPARQL to exploit the semantic metadata embedded in the WSDL contents of Web services. Thus, it is interesting to provide them intuitive and easily approachable ways to discover, to compose or to classify Web services.

Inspired by the dynamics of Web 2.0, we encourage user communities to participate in sharing their perceptions on Web services by tagging collaboratively any Web service that they used or tested before. Only through real world testing and usage, a user can have a clear idea on how to comment or give feedback on a Web service. Service usage experience can be expressed as comments or feedbacks on several aspects of Web service that can be observed by users during its testing or usage. We are interested in bridging the gap between original description and real world usage of service.

In this research work, we propose a folksonomic annotation model allowing users to express their perception on service functionality (after testing or using them). The annotations issued from users form incrementally community-based semantics on annotated Web services. This semantic metadata can be viewed by other users of the system and can be used to assist them in the task of discovery, composition or clustering of services. We propose also a collaborative tagging system supporting the aforementioned annotation model and allowing collect and construct social semantics of Web services.

This paper is organized as follows. Section 2 discusses similar research works. Section 3 describes the annotation/tagging model followed by the supporting tagging system in section 4. Section 5 argues on some applicative usage scenarios of the annotation to prove the pragmatism of our approach. Finally we conclude and present some future directions of the proposal.

II. RELATED WORKS

To our knowledge, there are still few existing proposals that adhere to the idea of using community-based semantics of Web services. We choose some significant proposals that follow the concepts and try to point out their strengths and limits.

The earliest proposal to use tagging to represent the semantics of Web service is discussed in [14]. It is still a rudimentary approach by considering keywords provided by users and associated to Web services as sufficient to describe their semantics. It is a collaborative tagging process but there is no discussion on how to merge tags provided by different users to annotate a specific service. Later on, a similar approach follows the same tagging mechanism but improves the system with more complete tags management in terms of variety and aggregation [15]. However, it shares with the previous one the same downside. Despite all, this concept of user-provided semantics of Web service is exploited in the real world situation in [16] and [17] to help users in searching for services.

Some improvement of the concept has been made in [18] by tagging only the *input* and *output* parts of Web services with the final goal to facilitate the discovery for composition process. However, the main drawback of the approach is that the original description of Web services is modified, which can be done only by the providers. Tags are then used to build a hierarchy with help from a machine learning approach, which requires a great quantity of annotated services and time. The discovery process using tags is based on this hierarchy. This idea of tag hierarchy is also picked up by [19] with help from WordNet [20] and Wikipedia [21] by exploiting their linguistic relations between terms. This hierarchy is also served to facilitate the discovery by tag. Nonetheless, both approaches ignore how tags are merged when provided by several users and associated to a service and still consider freeform tags as representative enough to reflect the service semantics.

In brief, there are common limits of the aforementioned proposals such as:

- Considering service semantics as global and taking freeform tags that are associated to the service as sufficient semantic expression of the service. This concept is still very vague because the semantics of service should be expressed in a more elaborated way by favoring more expression on the capability of the services
- Several users can tag a service but their aggregations are usually ignored. In fact, in a collaborative tagging system tag weight or tag frequency is an important factor to assess the significance of the tagged object via that tag and
- Tag ambiguity is often ignored. This issue can cause very weak ratio of *precision* and *recall* in the search of tagged objects by using tags.

Our proposal attempts to provide some solutions to improve the aforementioned limits encountered in the existing approaches.

III. ANNOTATION/TAGGING MODEL

It is worth mentioning that we work with publicly accessible Web services that can be collected from various Web service portals or catalogs on the Internet. Therefore, users are not allowed to add services into the system. Only the service providers can modify the service collection. Users can only add metadata to annotate or tag the existing Web services in the collection.

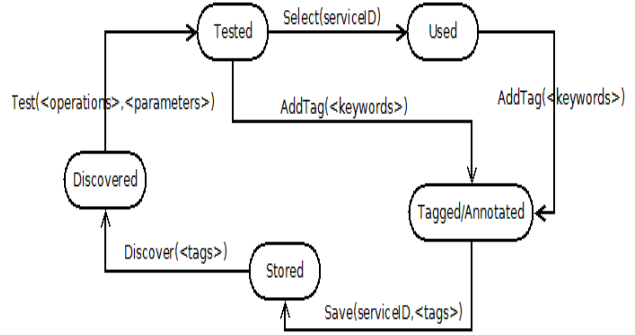


Figure 1. Web service annotation/tagging cycle

Generally, a public Web service can be discovered, tested, tagged or annotated (in our case) and used (e.g. in a composition). This cycle can be repeated as many times as possible and conducted by any user on any service. Thus, a public Web service can be present in various states of the aforementioned cycle (see Figure 1). At the user side, in order to get a full and clear idea on the functionality of a Web service, testing it is strongly recommended. Testing a service generally refers to testing the operations of the service by providing input data and observing output data. Therefore, at this state, the functionality of a Web service is qualified by its operations and for each of them what input and output is required (see the unit test interface of Seekda [17] and Directory of public SOAP Web services [22]).

However, a service can be employed differently with respect to usage context or user requirement (e.g. a weather forecast service can be used perfectly in either “holiday preparation” or “wedding party organization” context). Thus, operational and contextual information are necessary to fully describe the user perception on a Web service after a usage experience of the service.

In brief, given a Web service described by WSDL content, there can be several operations to test and for each operation there can be several input parameters to enter and output results to observe. Consequently, a tester can annotate a Web service at usage (describing the situation or context in which he/she uses the service), operation (describing different atomic functions of the service), input (describing what to offer as input when testing an operation) and output (describing the observable result after testing an operation) level.

Therefore, given a service, we can identify four types of *taggable object* that correspond to the four aforementioned levels (see Figure 2). We ignore the pre-condition and effect at operation level because a unit testing is done completely stateless and these aspects are hardly perceived by users.

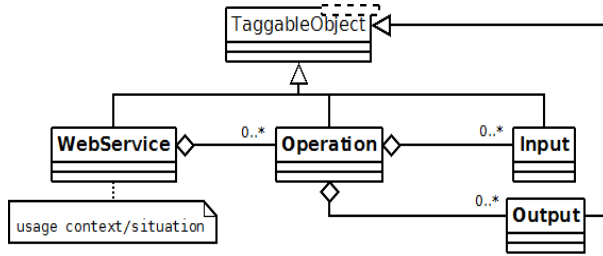


Figure 2. Web service annotation/tagging level

A user can annotated a Web service as efficient to be used in a certain context or situation, as providing several functions and each of them executes an action or as requiring some input data and providing certain output result.

The system allows collaborative participation of users in the tagging or annotation activities. In effect, a *taggable object* can be annotated by several users with different keywords and a user can annotate many *taggable objects* as well. Keywords from several users are used to construct tags and then feed the tag cloud that is associated to the corresponding *taggable object* (see Figure 3).

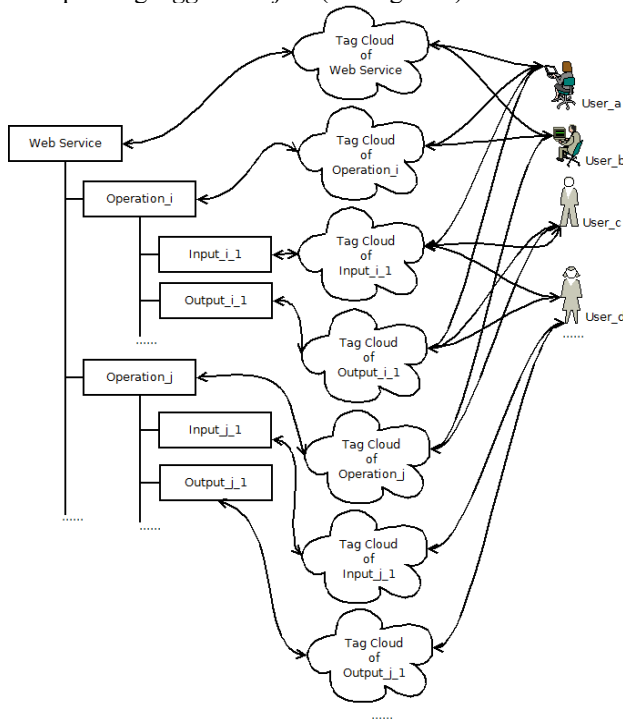


Figure 3. An instance of folksonomy structure

The annotations from a user on a service can be viewed by other users of the system in order to insure the community-based collaborative criterion.

Therefore, we can define our annotation model as a folksonomy $F \subseteq (O \times T \times U)$ where O , T and U are distinct sets and U : represents the set of users subscribed to the system and capable of annotating and tagging the Web service collection provided by the system, O : represents the set of objects that can be tagged (*taggable object*) by users. Each object can be a Web service (service usage context or

situation), an operation of a Web service, an input of an operation, or an output of an operation and T : represents the set of tags and each of them is a couple of a name label (keyword) k and a meaning $m \langle k, m \rangle$ that distinguishes the related keyword from another.

The next section discusses in detail the tagging system and describes its core components. We attempt to illustrate the functionality of such system by presenting the abstract architecture of the system and its interactions with users.

IV. ANNOTATION/TAGGING SYSTEM

After choosing a Web service to tag, the identifier of the service (its invocation URL) is input to the system. The system parses the WSDL file (with help from *WSDL Parser* module – (see Figure 4)) corresponding to the description of Web service in order to display its interfaces, operations, inputs and outputs. This parsing process is needed to visualize the Web service as structurally perceived by the user, mentioned in Figure 2.

A WSDL file can offer x *taggable objects* with $x=1+\text{number of operations} + \text{number of inputs} + \text{number of outputs}$. An *operation*, *input*, *output object* is identified by the Web service identifier and its XPath path (including namespace) of the corresponding element in the WSDL file.

After the parsing, the system searches for the existing *tag cloud* that is associated to each *taggable object* and displays them to users. This can give the user some ideas of how the others perceive the service he/she is tagging. Partial tagging is completely allowed in order to diminish the user effort in interacting with the system; i.e. the user is not obliged to tag every single item of the Web service.

Then the user can proceed to the effective tagging by providing a list of keywords to be associated to a chosen object. Each keyword is delimited by “” and separated by ;. We recommend users to provide dictionary vocabularies instead of personalized terms as keywords. In effect, personalized terms (e.g. nyc for New York City, fb for Facebook, etc.) can represent an important source of littering tags for the global tag cloud, thereby affects the quality of search operations using tags.

Each keyword can be associated to a set of meanings proposed by the system. The user chooses a meaning that suits best his/her keyword to form a tag. In effect, there are several linguistic issues related to collaborative tagging systems mainly *synonymy* and *polysemy* [23]. We decide to treat only polysemous keywords in our system because it represents an important element judging the *precision* rate of discovery/search system based on tags. This issue goes against our initial objective that proposes to use collaborative tags to enhance Web service discovery, reuse and composition processes, where *precision* rate is important for the system performance.

As soon as the keywords are semantified and the tags are formed, they are aggregated with the existing ones, given a *taggable object*. We opt for *bag-of-words* aggregation method that consists of increasing the frequency of tags in the tag cloud. A recent research proves that tag frequency is significant to qualify the tagged object by the related tag

because in most of the cases, tag distribution follows a power law [24].

The result of the aggregation is another *tag cloud* combining the tags provided by the user and those associated to the tagged object before the tagging. Finally, the tagged object and its *tag cloud* are saved back into the system (see Figure 4).

In detail, given a Web service identifier (its invocation URL), the *Tagged Web Service Collection* can output the corresponding WSDL file using its *Access Interface*. All the existing tags associated with this Web service are output by the same interface also.

In this collection, we store different instances of our folksonomy represented by 3 relations (see Figure 5) such as *User-Tag*: given a user u and a tag t , this relation offers the list of objects tagged by t . For each object, it indicates also the number of times (weight or frequency) t is used by u to tag the object, *Object-Tag*: given an object o and a tag t , this relation offer the list of users employing t to tag o . For each user, it indicates number of times he/she has used t to tag o and *User-Object*: given a user u and an object o , this relation offers the list of tags used by u to tag o . For each tag, it indicates its frequency also.

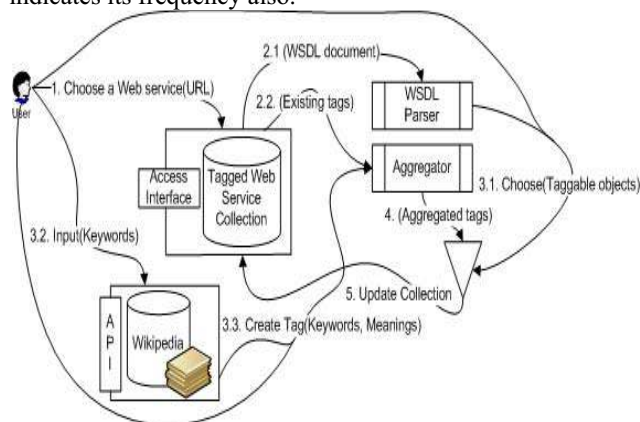


Figure 4. User-System modules interaction while tagging

WSDL Parser parses the description WSDL of the service to tag by proposing a structure of *taggable objects*. Users can then choose some objects to tag.

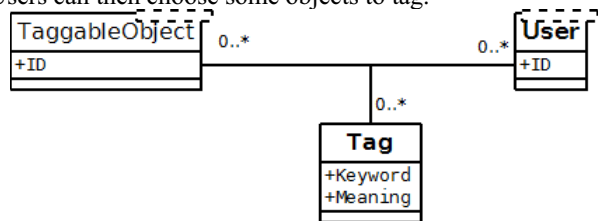


Figure 5. Taggable object - Tag - User relations

First, tagging an object consists of fetching input keywords from users and for each keyword, the system checks through Wikipedia [21] for its various meanings. This operation is called *disambiguation of keyword*. It is simply done by inquiring for the *disambiguation* pages of Wikipedia using some API such as [25] and [26]. Each *disambiguation*

page corresponds to the meaning that users chose to qualify a keyword.

In effect, the choice for Wikipedia is because it is a collaboratively edited free encyclopedia that evolves with strong support from community and covers a large number of vocabularies. It is possible that users leave the meaning of a keyword as *null* if none of the meanings proposed by Wikipedia suits him/her.

Secondly, after the meanings are selected by users, the tags are formed; i.e. couples of keywords and meanings. These tags are then aggregated with the existing tag cloud associated to the tagged object. The aggregation of two tag $t_i < k_i, m_i >$ and $t_j < k_j, m_j >$ having respectively f_i and f_j as frequency is a bag of tags (*tag cloud*) containing t_i if and only if t_i and t_j are identical (keyword $k_i = k_j$ and meaning $m_i = m_j$) with the new frequency that is equal to $(f_i + f_j)$ (one tag only); else, t_i and t_j with their previous frequencies respectively (two tags). After associating the aggregated tag cloud to the tagged object, the system updates the *Tagged Web Service Collection* by updating its three relations mentioned before, respectively.

V. SOME APPLICATIVE USAGE SCENARIOS USING OUR ANNOTATION/TAGGING MODEL

The fact that users can annotate Web services with their own tags implies more plentiful and community supported semantics. Beyond syntactic information, these semantics can enhance the discoverability, reusability and composability of Web services.

During the Web service discovery process at design time, our annotation can be used to help application developers in finding relevant capabilities of services. In this case, tags at Web service (usage context) and operation level are considered because they are significant to get the functional offer of the services. Web services can also be browsed via tags and that is more intuitive for developer to create his/her discovery query easily, quickly and efficiently. However, it is recommended that service user communities care enough participating in tagging Web services at the two aforementioned levels to make social semantics of the services emerge.

Web service composition can be assisted by users also. In this case, tags at input and output levels are more considered than those of usage context or operation level. When a user would like to have on-time effects or interventions on the composition of Web services, he/she can check the tag clouds at his/her disposal for atomic service that responds to his/her prompt need in order to create his/her composite Web service. The requirement formulation is easy as our tagging model favors the annotation on input and output level of Web services.

Tags themselves can be used to create user profile and propose the recommendation to help in the discovery and composition process. User profile may be constructed from tag sets offered by the user and be used in the filtering phase of the discovery process whereas recommendation can be used in the composition process to enhance the composability of the service.

Moreover, our annotation can help in the user-oriented and semi-automatic Web service clustering or classification also. With the increasing number of tags at operation levels, we can employ different natural language processing techniques on these tags to classify the tagged Web services.

Despite all, there are several challenges to overcome in order to evaluate our annotation model. Our system demands user participation to populate the tag clouds and their volume are important to access the quality and performance of the model. This factor is common to other collaborative tagging system as mentioned in [24] and [23]. Besides, it is even harder to conduct a comparative qualitative evaluation on our because of the weak number of existing semantic annotation models of the kind and their supporting systems.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed an annotation model that is easy to use and issued from the user perception on the functionality of Web services after testing or using them. The annotation model covers four levels of Web services: *usage context, operation, input and output*. Elements extracted from WSDL descriptions of Web services corresponding to each level represent system *taggable/annotatable objects*.

Several users can tag/annotate several objects during his/her system use. The tagging allows users to provide a list of keywords associated to an object and the system proposes several meaning of the keywords before asking the taggers to validate the suitable meanings. Thus, a tag is represented by a name entity and a meaning that is richer and unambiguous comparing to traditional freeform tags. The disambiguation of name entities or keywords is based the *disambiguation functions/pages* of Wikipedia which is the largest community supported and evolving encyclopedia.

The frequency of tag or tag weight evolves through user tagging activities and is essential to qualify the significance or semantic of the tagged object via the keywords.

We propose some applicative usage scenarios in which our annotation model can contribute in enhancing the overall performance and facilitating user interventions. It is also important to note that our annotation model does not modify the original descriptions of Web services and it is very plausible to use this model with the ontology-based semantic annotations presented in section 1. Both approaches are complementary because ours is more focused on the emerging semantics from the user side and those presented in section 1 are more used at the service provider sides.

In our future work, we would like to extend our annotation model to cover other semantic aspects of Web services such as non-functional or behavioral characteristics and use or adapt it in RESTful service context. We aim also at making use of existing SWS annotations by lowering them into human readable keyword meanings and employ them in the keyword disambiguation phase. We further provide multiple meaning bases to users allowing them to enrich their tag meanings and offer users the possibilities to add tailored meaning to their tags if none of the ones proposed by the system suits.

REFERENCES

- [1] "OWL-S: Semantic Markup for Web Services" Available: <http://www.w3.org/Submission/OWL-S/>.
- [2] "Web Service Modeling Ontology (WSMO)" Available: <http://www.w3.org/Submission/WSMO/>.
- [3] "Web Service Semantics - WSDL-S" Available: <http://www.w3.org/Submission/WSDL-S/>.
- [4] "Semantic Annotations for WSDL and XML Schema" Available: <http://www.w3.org/TR/sawsdl/>.
- [5] "D11v0.2 WSMO-Lite" Available: <http://www.wsmo.org/TR/d11/v0.2/>.
- [6] X. Shi, "Semantic Web Services: An Unfulfilled Promise," *IEEE IT Professional*, vol. 9, 2007, pp. 42-45.
- [7] D. Martin, J. Domingue, M.L. Brodie, and F. Leymann, "Semantic Web Services, Part 1," *IEEE Intelligent Systems*, vol. 22, 2007, pp. 12-17.
- [8] D. Martin, J. Domingue, A. Sheth, S. Battle, K. Sycara, and D. Fensel, "Semantic Web Services, Part 2," *IEEE Intelligent Systems*, vol. 22, 2007, pp. 8-15.
- [9] C. Schroth and T. Janner, "Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services," *IEEE IT Professional*, vol. 9, 2007, pp. 36-41.
- [10] C. Schroth, "Web 2.0 versus SOA: Converging Concepts Enabling Seamless Cross-Organizational Collaboration," *IEEE International Conference on E-Commerce Technology and Enterprise Computing, E-Commerce, and E-Services*, IEEE Computer Society, 2007, pp. 47-54.
- [11] C. Wu and E. Chang, "Searching Services "on the Web": A Public Web Services Discovery Approach," *2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System*, Shanghai, China: 2007, pp. 321-328.
- [12] S. Hageman, C. Letz, and G. Vossen, *Web Service Discovery - Reality Check 2.0*.
- [13] E. Al-Masri and Q.H. Mahmoud, "Discovering Web Services in Search Engines," *IEEE Internet Computing*, vol. 12, 2008, pp. 74-77.
- [14] H. Meyer and M. Weske, "Light-Weight Semantic Service Annotations Through Tagging," *Service-Oriented Computing - ICSOC 2006*, 2006, pp. 465-470.
- [15] U. Chukmol, A. Benharkat and Y. Amghar, "Enhancing Web Service Discovery by Using Collaborative Tagging System," *International Conference on Next Generation Web Services Practices*, IEEE Computer Society, 2008, pp. 54-59.
- [16] M.K. Papathanasiou, N. Loutas, V. Peristeras, and K. Tarampanis, "Combining Service Models, Semantic and Web 2.0 Technologies to Create a Rich Citizen Experience," *Proceedings of the 2nd World Summit on the Knowledge Society: Visioning and Engineering the Knowledge Society. A Web Science Perspective*, Chania, Crete, Greece: Springer-Verlag, 2009, pp. 296-305.
- [17] "seekda! - About seekda" Available: <http://webservices.seekda.com/>.
- [18] E. Bouillet, M. Feblowitz, H. Feng, Z. Liu, A. Ranganathan and A. Riabov, "A Folksonomy-Based Model of Web Services for Discovery and Automatic Composition," *IEEE International Conference on Services Computing*, IEEE Computer Society, 2008, pp. 389-396.
- [19] I. Silva-Lepe, R. Subramanian, I. Rouvellou, T. Mikalsen, J. Diament and A. Iyengar, "SOALive Service Catalog: A Simplified Approach to Describing, Discovering and Composing Situational Enterprise Services," *Proceedings of the 6th International Conference on Service-Oriented Computing*, Sydney, Australia: Springer-Verlag, 2008, pp. 422-437.
- [20] "About WordNet - WordNet - About WordNet" Available: <http://wordnet.princeton.edu/>.
- [21] "Wikipedia, the free encyclopedia" Available: http://en.wikipedia.org/wiki/Main_Page.

- [22] "Directory of public SOAP Web Services" Available: <http://www.service-repository.com/>.
- [23] S.A. Golder and B.A. Huberman, "The Structure of Collaborative Tagging Systems," 2005.
- [24] V. Robu, H. Halpin, and H. Shepherd, "Emergence of consensus and shared vocabularies in collaborative tagging systems," *ACM Trans. Web*, vol. 3, 2009, pp. 1-34.
- [25] "JWPL" Available: <http://www.ukp.tu-darmstadt.de/software/jwpl/>.
- [26] "Wikipedia API - Wikipedia-Lab" Available: http://wikipedia-lab.org/en/index.php/Wikipedia_API.