# A MOTIVATION FOR A MODERN WEB SERVICE DESCRIPTION

Konstanty Haniewicz
*Poznan University of Economics, al. Niepodległości 10, Poznań, Poland*

Keywords: Web services, Web service description, Service Oriented Architecture, SOA oriented enterprises.

Abstract: The aim of this work is to present audience with a motivation for research aimed at introducing a new approach towards a Web service description. The description is referenced mainly at Service Oriented Architecture oriented enterprises, as they are the most plausible candidates for its employment. Nevertheless, one might perceive it as a proposition of changes that could have wider reception. The cornerstone for the research is based on available solutions for a Web service description that are contrasted with requirements arising from business practitioners. Carried research allowed for stating a set of requirements that are to be met by a solution that shall improve process of a Web service retrieval. This works concludes with observations and postulates concerning a modern Web service description.

## 1 INTRODUCTION

It is believed that Web services are the most important tool in implementation of Service Oriented Architecture. At this point every major production-scale framework for software production supports Web services. One might risk stating that Web services became a de-facto standard for interoperability.

Out of all standards created to enable Web services, the most important ones are the Web Service Description Language (WSDL), the Universal Data Discovery and Integration (UDDI) and various others devoted to security, quality and other aspects (involving those that do not abide the WSDL hegemony). Not all of the crafted standards became widely adopted by the industry, yet the two explicitly enumerated have greatly influenced other initiatives.

Aside from the industry strategies, Web services became an area of interest for the academia. Very soon, research works departed from the research of interoperability to ignite whole new interest in other features available thanks to employment of Web services. Researchers observed a great potential of Web services as an abstraction for certain routines that could be used to produce highly configurable software. To enable this to happen, a number of challenges was identified (Papazoglou, 2007).

First of all, one has to be able to manage his Web service repository. A considerable number of works tackled the problem pointing out that the initial solution (UDDI) is anything but sufficient (Klein, 2004). Addressing various deficiencies of UDDI, researchers proposed a number of enhancements (Hicks, 2007). These enhancements were realised by building systems that could use introduced features provided by additions to original WSDL document. Additions to WSDL documents could come from semantic extensions (Paolucci, 2002), inclusion of description logic elements not provided by one of the specific semantic languages (Colucci, 2003) or processing of WSDL documents and using the results in the envisioned systems (Al-Masri, 2009). Over the course of years, semantic additions became the most prolific area in the domain of Web services. One has to clearly state that, popularity of semantic extensions was yielded by the notion of Semantic net and the idea of automatic composition of software and data obtained by enrichment of both, with metadata and introduction of mechanisms capable of reasoning over it (Vitvar, 2007). Application of semantics resulted in rise of Semantic Web Services (McIlraith, 2001). For a few years SWS become a term almost universally interchangeable with Web services.

Haniewicz K..
A MOTIVATION FOR A MODERN WEB SERVICE DESCRIPTION.
DOI: 10.5220/0003609801550158
In *Proceedings of the International Conference on e-Business* (ICE-B-2011), pages 155-158
ISBN: 978-989-8425-70-6

## 2 CHALLENGES OF A WEB SERVICE DESCRIPTION

There are a number of approaches to Web service retrieval. As reported earlier, Web service is in essence an abstraction for a routine implemented in a programming language. At this point, one has to explicitly underline that in most of industry strength solutions provided by companies such as IBM, Microsoft, Oracle or SAP a Web service is nothing more than this abstraction at a level of WSDL document.

Therefore, at least two main approaches were crafted. The first one is based mainly on Web service operation signatures. Main methods of this approach focus on matching of Web service operations' signatures. This resembles activities of earliest research concerned with software components reuse (Mili, 1997). Signature is not sufficient to identify purpose and actual activities of any procedure. Web services were to address this issue by taking into account preconditions and effects. Yet, both were broadly disregarded by Web service community. The sources of disregard are beyond the scope of this work, nevertheless, one can reason that its aim were not clear enough to users or there was not sufficient amount of support at deployment level.

The second approach takes the observation on insufficient data stored in routine signature to safely determine its purpose and uses it as a foundation for employing any additional data available. This additional data is stored both in comments inside routine (and in its neighbourhood) and as actual names for routine and its parameters.

Both solutions are sure to fail when applied to a large corpus of Web services (Petrie, 2009). Therefore, some additional actions must be undertaken in order to manage data overload. Organizations that would like to benefit from tools based on any of further discussed groups of solutions need to face one or more of following challenges:

- Semantic information must be attached to Web services stored in repository
- semantic data must be processed in order to resolve user queries
- Web services represented only by WSDL documents thus being signatures are not sufficient for effective retrieval
- Precise results from queries on Semantic Web services induce cost of ontology preparation, weakly prepared ontology results in low efficiency of query's output
- Extra costs of documenting all items held in repository, documentation from source code of abstracted routines might not apply to Web service itself
- Tagging content with simple categories is too general and does not save users from further retrieval by scrutiny
- Cost of search for desired functionality affects decisions on reimplementation.

Taking into account the above, domain literature review and results of preliminary interviews with business practitioners working in IT departments, there are following postulates that a solution for Web service retrieval and representation should have.

Desired solution shall respect current philosophy employed in design of interfaces search engines. This is important as users will not be exposed to another learning curve and will allow for streamlined adoption among them.

Another important feature that has to be addressed is description complexity of a single Web service and additional resources compulsory for efficient retrieval. Every additional description element increases complexity. When complexity is high, provided data can be incorrect due to mistakes or negligence. Not to mention increased effort of users generated costs of skill acquisition and cost of description itself.

Precision is another required feature. Results must match queries and prevent from presenting user with abundant information.

Retrieval process shall consume little amount of time, as business requirements emphasize fast response. Moreover, interface shall be snappy and responsive. Solution shall be ready for handling of hundreds of thousands of Web services.

Performance cannot degrade significantly with addition of new Web services. Next section shall cover the available solutions proposed with the academia. Before that, one has to underline the fact that a perfect solution shall enable its user to fully exploit all advantages provided by Web services for Service Oriented Architecture.

Moreover, anyone that shall try to present a modern Web service description shall respond to following questions:

- How one should represent Web service in order to achieve high precision and manage description costs at a low level?
- How one should design a system taking into account all postulates in order to protect it from performance degradation while increasing number of stored Web services?

## 3 SUMMARY OF A WEB SERVICE DESCRIPTION SOLUTIONS

In general, for over 10 years there is no consensus whether Web services should be enhanced with semantic annotations or some other method that increase effectiveness of retrieval should be applied. Solutions proposed range from simple retrieval base on processed keywords available in WSDL documents to those that not only use languages such as WSML and OWL for description (Maigre, 2010) of basic functionality but also expression of other desired by users features.

Most prominent trend in Web service description is semantic annotation. Greatest benefit of description with ontology is ability to drastically increase precision and recall of retrieval process. It's achieved by unprecedented power to express a model of some world. Unfortunately, this rebounds at level of complexity for both world description builders and model end users.

On the other end of spectrum, there are description methods that delve into terms used in WSDL documents (Wu, 2005). In its most naive form they propose indexing of all available terms and presenting results as ranked list of WSDL documents that match query terms. More advanced solutions allow for pre-processing of terms in order to filter out possible synonyms and ambiguities. It was proven that it yields better results than naive approach. Using traditional retrieval can be perceived as an iterative process as previous improvements gain recognition and are included in later works. Thus when filtering of synonyms, and partial disambiguation of terms became established technique, it was enriched by attempts to build ontology of concepts available in corpus of Web services accessible to researchers. As a drawback, one has to highlight that from this point on any retrieval had to be aided by human operator to some extent.

Few researchers decided to describe Web services in alternative manner. Most interesting method devised is capturing Web service as a pattern of states and transitions between them (Rocco, 2005). Flow graphs enabled to perceive what one can do with given Web service as he is presented with a list of viable possibilities. One has to underline that these attempts although tempting, cannot relieve Web service descriptions builders and end users from effort spent in learning how to efficiently model a Web service and later retrieve it.

One last category is hybrid approaches that do not focus on Web service description technology per se, but on its scope. This is the most varied category by far. Common denominator for its members is inclusion of features that are ignored by previous categories. As Quality of Service and non-functional properties are handled in some solutions (WSMO recognizes non-functional parameters, it realizes some of QoS with them) distinctive features come from recognition and addressing issues such as multiple perspective of Web service, Service Level Agreement bound to specific operations and Web services, trust and ability to resolve fuzzily stated requirements (Cardoso, 2010).

It is crucial, to once more highlight that the technology in which Web services are described is of secondary importance. The key, are the features provided.

## 4 CONCLUSIONS

One could generalize available solutions in a manner that varies from those proposed in (Dmello, 2010) due to the fact that a business user shall not differentiate between solutions oriented on functionality or non-functional parameters. They want a functional entity that shall empower them to realise their business objectives with minimal effort. Therefore, there is a number of postulates for a modern Web service description.

First of all, one would need a purpose statement of every Web service. This is realized by a few solutions, one of them is WSMO which allows for Web service goals. Nevertheless, these service goals are expressed as capabilities addressing every element from IOPE quartet. This cannot be universal solution as a business user is not interested in preconditions and post conditions. He is interested in finding a Web service that brings concrete results and he wants to find it without extra effort in analyzing ontology interdependencies. It was observed that tagging systems, based on some taxonomy are of great interest as they aid to fill these needs. A Web service is categorized not with some unrelated terms, but with terms coming from business user environment.

Thus, Web service description shall take into account its context. Without it, it is yet another abstraction layer that can find application possibly for developers when it's documented. By Web service context, one can understand its application in organization. Why was it prepared, and in what terms it was documented. Finally, how it wasclassified by its builders.

More, Web service can mean different things for different users. Business user would like to acquire a building block that can be employed into his business process and enable him to produce added value. Developer would like to be able to locate Web services that can implement some desired functionality in order to save him from unnecessary work that could be invested elsewhere. Architect of organization's system would like to audit state of affairs and quickly asses whether some functionality is under or over represented and act accordingly. External contractor would like to quickly check whether he can introduce some functionality so that it can find application in organization and thus bring him revenue.

Good Web service description shall allow for multidimensional tagging with a number of taxonomies. One cannot believe that these taxonomies should be built automatically. The process can be aided by traversal of available documentation and additional input from users. Ultimately, every taxonomy must be prepared by a skilled user. Yet when prepared for some compact area it shall be still easily comprehensible for users unlike oversized ontologies striving for depicting domain exhaustively.

One shall believe that only a solution that is able to harness complexity of semantic technologies and combines it with good practices taken from WSDL oriented retrieval can be truly successful.

## REFERENCES

Papazoglou M. P, Traverso P., Dustdar S., Leymann F., Service-Oriented Computing: State of the Art and Research Challenges, Computer, vol. 40, no. 11, pp. 38-45, Oct. 2007, doi:10.1109/MC.2007.400

D'Mello D. A., Ananthanarayana V.S., A Review of Dynamic Web Service Description and Discovery Techniques, iciic, pp.246-251, 2010 First International Conference on Integrated Intelligent Computing, 2010

Web Services Description Language (WSDL) v. 2.0, World Wide Web Consortium (W3C) recommendation, June 2007; www. w3.org/TR/wsdl20.

Vitvar T., et al., Semantically-Enabled Service-Oriented Architecture: Concepts, Technology and Application, Service Oriented Computing and Applications, vol. 2, no. 2, 2007, pp. 129–154.

UDDI Version 3.0.1, available at http://uddi.org/pubs/ uddi-v3.0.1-20031014.htm

Petrie C., Practical Web Services. *IEEE Internet Computing* 13, 6 (November 2009), 93-96.

McIlraith S. A., Son T. C., Zeng H.. 2001. Semantic Web Services. *IEEE Intelligent Systems* 16, 2 (March 2001), 46-53.

Al-Masri E., Mahmoud Q.H., Discovering the best web service: A neural network-based solution, *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, IEEE, 2009, p. 4250–4255.

Crasso M., Rodriguez J.M., Zunino A., and Campo, M. Revising WSDL Documents: Why and How, *IEEE Internet Computing*, vol. 14, Sep. 2010, pp. 48-56.

Hicks J., Govindaraju M., and Meng W., Search Algorithms for Discovery of Web Services, *IEEE International Conference on Web Services (ICWS 2007)*, Jul. 2007, pp. 1172-1173.

Klein M. and Bernstein A., Toward high-precision service retrieval, *Internet Computing, IEEE*, vol. 8, 2004, p. 30–36.

Mili R., Mili A., and Mittermeir R.T., Storing and Retrieving Software Components : A Refinement Based System, vol. 23, 1997, pp. 445-460.

Paolucci M., Kawamura T., Payne T., and Sycara K., Importing the semantic web in uddi, *Web Services, E-Business, and the Semantic Web*, 2002, p. 815–821.

Colucci S., Di Noia T., Di Sciascio E., Donini F.M., and Mongiello M., Description Logics Approach to Semantic Matching of Web Services, *Journal of Computing and Information Technology*, vol. 11, 2003, pp. 217-224.

Cardoso J., Barros A., May N., and Kylau U., Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments, *Services Computing (SCC), 2010 IEEE International Conference on*, IEEE, 2010, p. 602–609.

Maigre R., Survey of the Tools for Automating Service Composition, *2010 IEEE International Conference on Web Services*, Jul. 2010, pp. 628-629.

Wu J., Similarity-based Web Service Matchmaking, *2005 IEEE International Conference on Services Computing (SCC’05) Vol-1*, 2005, pp. 287-294.

Rocco D., Caverlee J., and Critchlow T., Domain-Specific Web Service Discovery with Service Class Descriptions, *IEEE International Conference on Web Services (ICWS’05)*, pp. 481-488.