Light-Weight Semantic Service Annotations Through Tagging

Harald Meyer and Mathias Weske

Hasso-Plattner-Institute for IT-Systems-Engineering at the University of Potsdam
Prof.-Dr.-Helmert-Strasse 2-3, 14482 Potsdam, Germany
{harald.meyer, mathias.weske}@hpi.uni-potsdam.de

Abstract. Discovering and composing services is a core functionality of a service-oriented software system. Semantic web services promise to support and (partially) automate these tasks. But creating semantic service specifications is a difficult, time-consuming, and error prone task which is typically performed by service engineers. In this paper, we present a community-based approach to the creation of semantic service specifications. Inspired by concepts from emergent semantics and folksonomies, we introduce semantic service specifications with restricted expressiveness. Instead of capturing service functionality through preconditions and effects, services are tagged with categories. An example illustrates the pragmatic nature of our approach in comparison to existing approaches.

1 Introduction

The goal of service orientation is the alignment of the IT infrastructure to the business goals of a company [1,2,3]. Service-oriented architecture (SOA) defines the elements and relations of such an IT infrastructure. Two of the core tasks in a SOA are discovering services and composing services into new services to fulfill complex tasks [4,5]. In the presence of hundreds or thousands of services, both tasks become challenging. Semantic web services [6] are a promising approach to find services based on functionality. Service functionality is described through preconditions and effects. Creating them and writing queries to find services according to preconditions and effects is a complex task.

In this paper, we present a novel approach towards service semantics for service discovery and composition. Instead of assuming fully automated discovery and composition, we want to assist users with these tasks. For this, preconditions and effects are not necessary. Service users (process designers, etc.) tag services with keywords. These tags enable them to find services. While a service engineer can provide an initial categorization for a service, users can refine categorizations incrementally. This helps capturing real world aspects of service usage and bridging the gap between service description and real world service usage.

The approach presented is similar to *service categories* in OWL-S [7] and WSDL-S [8]. But service categories are static. During development, the service engineer assigns suitable service categories to the new service. A strict separation between service development and service usage prevents changes by people

A. Dan and W. Lamersdorf (Eds.): ICSOC 2006, LNCS 4294, pp. 465–470, 2006.

other than the service engineer. Systems like NAICS (North American Industry Classification System), UNSPSC (United Nations Standard Products and Services Code), or RosettaNet have defined processes for changes to their taxonomies. Adding new concepts may take up to 5 years (for NAICS). Categories are statically assigned: it rarely makes sense to change a service categorization in UNSPSC from *4410260214* (Retail of Printers) to *4410310314* (Retail of Toners). Instead one would remove the existing service and publish a new one.

Our work is inspired by the recent advent of emergent semantics and folksonomies. Both approaches do not depend on a-priori semantical commitments. Instead, semantic commitment is achieved incrementally. In the next section we go into more details of emergent semantics and present a formalization for tags. This formalization serves as the foundation for our application of emergent semantics to service discovery in Section 3. The paper concludes with a summary and an outlook on future work.

2 Emergent Semantics

Technologies for semantic annotation originate in the annotation of documents. Recently, these technologies are also used to specify service functionality [6]. When annotating documents, annotations are created either by dedicated professionals or the authors of the documents [9,10]. Professionally created annotations are of high-quality, but their creation is costly and rarely scales for large amounts of documents. Author-created annotations overcome this problem. But in both approaches the actual users are detached from the creation of the annotations. Annotations might therefore not match the actual usage of the documents. If the usage of document changes or it is used in unintended ways, the annotations cannot reflect this.

Emergent semantics [11] replaces a-priori agreements by incremental, local agreements. The recent rise in *folksonomies* can be seen as an application of emergent semantics. The term *folksonomy* is a composition of folk and taxonomy. In a folksonomy annotations for documents are created by the users of the system through tagging. The most prominent examples for systems based on folksonomy are del.icio.us (`http://del.icio.us/`) and flickr (`http://flickr.com/`) bookmark and photo management systems. When adding a bookmark in del.icio.us you can add multiple tags or categories to the bookmark. Later, tags can be used to find the bookmark again. Another feature of folksonomies is their community orientation. Bookmarks and tags are shared among all users. Hence, a user cannot only find all the bookmarks he tagged with a given tag, but he can also find all the bookmarks tagged with the same tag by all other users.

The freedom resulting from the usage of tags leads to problems illustrated by Golder and Huberman [12]. It is possible that a tag has multiple homonymous or polysemous meanings. Synonyms can appear as well. Such synonyms can be especially complicated in a tagging-based system as user using one tag, will not find documents tagged with the other tag. User do not need to adhere to a naming convention. This is problematic as it is unclear whether a tag should be in singular (e.g. *book*) or in plural (e.g. *books*). How tags consisting of more than one words are composed is also undefined: write them as one word, separate them with underscore, or two tags. Similar problems occur on the level of message-level heterogeneities when Web services exchange data [13,14] and in multidatabase systems [15,16]. Michlmayer [17] identifies spamming as an additional problem. The assumption why these problems do not interfere with the actual usability of existing systems, is that most of them do not matter if only enough users participate. For example, the problems with synonyms is that two different tags for the same meaning lead to separated document landscapes. But if enough users participate, chances are high that most documents get tagged with both tags.

As a last point in this section we describe a formalization for emergent semantics introduced by Mika [18]. This formalization will later serve us as the basis for our formalization of emergent semantics for service annotation. A folksonomy:

Definition 1. *A folksonomy $F \subseteq A \times T \times O$ is hypergraph G(F)=(V,E) with*

- $V = A \cup T \cup O$ *as the vertices. The disjoint sets* $A = \{a_1, ..., a_k\}$, $T = \{t_1, ..., t_m\}$, $O = \{o_1, ..., o_n\}$ *are the set of actors, the set of tags, and the set of objects.*
- $E = \{\{a, t, o\} | (a, t, o) \in F\}$ *as the hyperedges connecting an actor a who tagged an object o with the tag t.*

3 Emergent Semantics for Service Annotation

To find services it is important to annotate them with a description of their functionality. Existing author-created or professionally created semantic annotations, are costly to produce and have the risk of not matching the actual usage of the service. Hence, we will apply the concept of tagging in this section as a light-weight approach towards semantic service annotations. As a first step we introduce service landscapes:

Definition 2. *A service is a discrete business functionality. It is described by a service description. A service landscape is the set of available services described by service descriptions* $\mathcal{S} = \{s_1, s_2, ..., s_n\}$.

In the upcoming semantic web service *standards* OWL-S [7] and WSMO [19] services are described through preconditions and effects. WSMO also introduces assumptions and postconditions. It distinguishes between information space and world space. SAWSDL [20] explicitly excludes "expression of Web services constraints and capabilities, including precondition and effect". The precondition defines if a service is invokable in the current state. A formal specification of the functional description of services can be found in [21]. If the precondition is satisfiable by the current state, the service is invokable. The effect describes the changes to the current state that result from invoking the service. With our approach preconditions and effects are no longer necessary. Instead tagging is applied to semantic services:

Definition 3. *A tagging-based semantic service system with a service landscape S is a folksonomy where the objects are the service landscape:* $F \subseteq A \times T \times S$.

This means service descriptions are tagged to express service functionality. The actors in such an environment are for example process designers, service landscape managers, and service engineers.

3.1 Example

This example will from now on serve as an illustration for our findings. The example is about leave requests by employees. Two different kinds of leave requests can be distinguished: vacation and sabbatical. Figure 1 shows the service landscape $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$. On the left side of each service the input parameters and on the right side the output parameters are denoted. The services s_1, s_2, and s_3 deal with vacation requests. After requesting a vacation (s_1), the request's validity (e.g. whether enough vacation days are left) is checked (s_2), and finally the vacation is approved or rejected (s_3).

Services s_4 and s_5 are the respective services for sabbatical requests. In contrast to vacation requests, no automated validity check is performed. Instead the supervisor needs to manually check the eligibility of the employee to go on a sabbatical. Finally, service s_6 is used to update the information about sabbaticals and vacations of employees in the human resources system. The human resources system then publishes the information to the project planning tools so that no work is planned for employees, who are on leave.

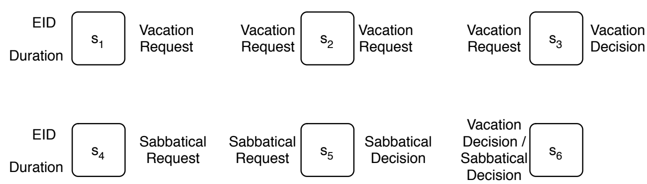

Fig. 1. Employee leave request: service landscape

Pete and Mary are process designers. Pete is the first one to model a process. He wants to model a process for vacation request approval. As no tags exist, he needs to browse the service landscape to find the required services s_1, s_2, s_3, and s_6. To help him and other persons in finding these services in the future, he tags them with the new tag *vacation*. This leads to the following folksonomy: $F = \{(Pete, vacation, s_1), (Pete, vacation, s_2), (Pete, vacation, s_3), (Pete, vacation, s_6)\}$. Mary works in another department that currently not tracks spent vacation days in the human resources system. Instead, the head of department uses a spreadsheet for this purpose. Hence, she does not need to use service s_2. As she sees Pete's *vacation* tag, she can easily figure out all useful services. The system does not store relations between services, so Mary has to model the process manually without the usage of s_2. While she found Pete's tags useful, she thinks fine granular tags are better. Hence, she introduces *vacation_request* to tag s_1, *vacation_approval* to tag s_3, and *update_leave_info* to tag s_6. She also models a new process for sabbatical requests using services s_4, s_5, and s_6. The folksonomy now is (Figure 2): $F = \{(Pete, vacation, s_1), (Pete, vacation, s_2), (Pete, vacation, s_3), (Pete, vacation, s_6), (Mary, vacation\_request, s_1), (Mary, vacation\_approval, s_3), (Mary, update\_leave\_info, s_6), (Mary, sabbatical\_request, s_4), (Mary, sabbatical\_approval, s_5)$.

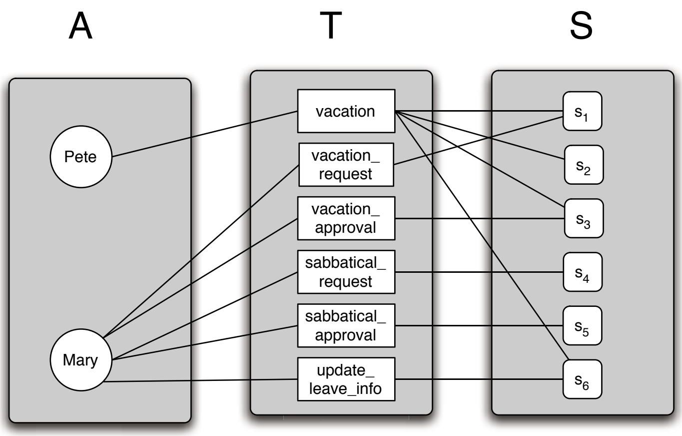

Fig. 2. Service Folksonomy

4 Conclusion

In this paper we presented a novel approach to service semantics. Instead of modeling service semantics up-front by service engineers, they are incrementally refined by the users. Existing Folksonomy implementations like del.icio.us are positive examples how the problems of community-based tagging can be solved. Systems like del.icio.us overcome these problems through their large user base. In comparison to several thousand users using such web-based system, we have to deal with a significantly smaller user base. In an intra-enterprise scenario maybe only a few dozen users use the system. While the user group is smaller, it is also of higher quality. Users have a direct gain in their daily work and have responsibility for their doings.

We already implemented a preliminary prototype for the displayed functionality. The next step is to integrate this functionality into an existing BPM suite and UDDI repository. As a part of this work, experiments will be conducted to prove the applicability and usefulness in real world process modeling.

References

1. Burbeck, S.: The tao of e-business services. IBM developerWorks (2000)
2. Papazoglou, M.P., Georgakopoulos, D.: Service-oriented computing: Introduction. Communications of the ACM **46** (2003) 24–28

3. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services – Concepts, Architectures and Applications. Data-Centric Systems and Applications. Springer (2004)
4. Curbera, F., Khalaf, R., Mukhi, N., Tai, S., Weerawarana, S.: The next step in web services. Communications of the ACM **46** (2003) 29–34
5. Milanovic, N., Malek, M.: Current solutions for web service composition. IEEE Internet Computing **8** (2004) 51–59
6. McIlraith, S.A., Son, T.C., Zeng, H.: Semantic web services. IEEE Intelligent Systems **16** (2001) 46–53
7. `http://www.daml.org/services/owl-s/1.0/`: OWL-S 1.0 Release. (2003)
8. `http://www.w3.org/Submission/WSDL-S/`: WSDL-S. (2005)
9. Rowley, J., Farrow, J.: Organizing Knowledge: Introduction to Access to Information. Gower Publishing Limited (2000)
10. Mathes, A.: Folksonomies - cooperative classification and communication through shared metadata. (2004)
11. Aberer, K., et al.: Emergent Semantics Principles and Issues. In: 9th International Conference on Database Systems for Advanced Applications. (2004) 25–38
12. Golder, S., Huberman, B.A.: The structure of collaborative tagging systems. Journal of Information Science (2005)
13. Sheth, A.P.: Changing focus on interoperability in information systems: From system, syntax, structure to semantics. In: Interoperating Geographic Information Systems, Kluwer Academic Publishers (1998) 5–30
14. Nagarajan, M., Verma, K., Sheth, A.P., Miller, J.A., Lathem, J.: Semantic interoperability of web services – challenges abd experiences. In: Proceedings of the 4th IEEE Intl. Conference on Web Services. (2006) (to appear).
15. Sheth, A.P., Kashyap, V.: So far (schematically) yet so near (semantically). In: Conference on Semantics of Interoperable Database Systems. (1992) 283–312
16. Kim, W., Choi, I., Gala, S.K., Scheevel, M.: On resolving schematic heterogeneity in multidatabase systems. Distributed and Parallel Databases **1** (1993) 251–279
17. Michlmayr, E.: A Case Study on Emergent Semantics in Communities. In: Proceedings of the Workshop on Social Network Analysis, International Semantic Web Comference (ISWC). (2005)
18. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Proceedings of the 4th International Semantic Web Conference (ISWC2005). Number 3729 in LNCS, Springer (2005) 522–536
19. `http://wsmo.org`: Web Service Modeling Ontology. (2005)
20. `http://www.w3.org/2002/ws/sawsdl/`: SAWSDL Working Group. (2006)
21. Keller, U., Lausen, H., Stollberg, M.: On the semantics of functional descriptions of web services. In: Proceedings of the 3rd European Semantic Web Conference (ESWC2006) (to appear). (2006)