

Chapter 50

Towards a Supporting Framework for Public Web Service Efficient Reuse: A Model Based on Social and Collective Usage Experience

Uddam Chukmol, Aïcha-Nabila Benharkat, and Youssef Amghar

1 Introduction

Public Web services are accessible and available Web services in the Internet. They are generally published within Web portals, catalogues or repositories directly managed by their providers (Fan and Kambhampati 2005). At the back end, service providers are not obliged to respect the UDDI standard for the implementation of their repositories (Wu and Chang 2007). Service providers offer to end users service lookup interface implemented within their repositories, and end users can invoke directly a found service from those repositories (AbuJarour et al. 2010).

Recent access to seekda¹ shows around 28,606 public Web services out of 7,739 providers, which is not negligible as a source of interest in promoting SOA adoption to a larger public than the one traditionally identified in intra-corporate environment. However, exploiting those publicly exposed Web services may come with some challenges due to the following facts:

- Poor textual documentation found in service description makes end users put more effort in understanding what a service can do, what it needs as input and what it can produce as output. Consumers may have to infer the operation names or message types used in the service in order to utilize them correctly (Li et al. 2007).
- Poor semantic extension reduces considerably the possibility to automate different essential tasks such as Web service discovery, composition and substitution. Semantic mark-up on Web services comes from Web semantic communities by extending original WSDL descriptions with various semantic formalisms based

¹ <http://webservices.seekda.com>

U. Chukmol (✉) • A.-N. Benharkat • Y. Amghar
Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
e-mail: uddam.chukmol@insa-lyon.fr; nabila.benharkat@insa-lyon.fr;
youssef.amghar@insa-lyon.fr

on ontology. Several available formalisms do not really facilitate the automation process but bring somehow burdens to both service providers, in choosing an appropriate formalism and domain ontology to use as semantic annotation, and consumers in acquiring expertise to construct discovery query or building business process schema for composing Web services (Hageman).

- Non-situational refers to the fact that public Web services are for general-purposed usage. They are not predestined to be used in a planned composition schema or any specific context. End users can consume them appropriately in their own ways which could be different from the original applicative aims targeted by service providers (Tanaka et al. 2008).

Difficult usage begets hard reuse. In effect, Web service reuse can be simply and practically put as several or repetitive usages of the same Web service instance through time and within different situations. The service is designed and published once, then used and reused over and over again by different consumers in different contexts. Web service reuse is strongly influenced by its usage and represents a great return on investment if it is well achieved.

Public Web services that are available “on the Web” provide unlimited values for a great number of online service consumers and developers, whereas most private registries would focus on a specific, closed domain such as the corporate network (Wu and Chang 2007). Hence, in an open, loosely coupled, demand-driven business environment, a maximal reuse of public Web services is still essential to SOA adoption from a wide range of users such as service consumers, service developers and service brokers.

In this chapter, we would like to tackle the reuse of publicly available Web services from the end-user point of view. In effect, after being published, end users discover, select and consume public Web services in their applications. They are then apt to comment such services on functional or non-functional aspect. Such feedbacks are recommended as supplementary semantics of Web services and can be served to enhance the *discoverability* and the *recommendability* of those services (Loutas et al. 2009; Petrie 2009; El-Goarany et al. 2008). Other users would consider if a service can be *reused* in their application based on the other usage experiences. We attempt in this chapter to provide a simple way by which users can express their service usage experiences. We outline a set of service *reusability* criteria and show how end-user feedback on the services can be used to satisfy those criteria and reinforce implicitly the service *reusability*.

This chapter is organized as follows: Sect. 2 discusses related works on Web service reusability and offers a synthesized set of criteria on Web service *reusability*. Section 3 proposes a formalism allowing end users to express their service usage experiences. Section 4 presents the architecture of a supporting system exploiting aforementioned feedbacks to promote and enhance Web service *reusability*. We end up with a conclusion and perspective on future works.

2 Related Works

Web service *reusability* is generally discussed as an issue in service design phase and rarely in consumption phase (published and ready to use). A main challenge is how to bridge the gap between Web services and general public. In effect, the benefit from reusable Web services helps consumers in getting better return on investment because the services can be employed in different situations in their applications and providers in attracting more consumers, gaining more credibility and being more trustworthy.

The existing works on reusable Web services are mostly focused on service providers rather than considering end-user perceptions on services. In Dan et al. (2008), reusing Web services refers to good governance in SOA lifecycle. The proposal is very enterprise oriented and consists of using a common language with consistent business term to describe services, governing new service creation and discovery of existing services, providing assurance to both service users and owners and supporting new requirements against existing services. Following the same idea, the proposal in Chu and Qian (2009) considers designing reuse-oriented Web service through a top-down process by modelling business requirement as business processes, breaking them down and identifying fine-grained requirement before the real service design. These two solutions may be applied efficiently in supervised and inter/intra-corporate environment where business terms can be controlled by internal glossary or domain ontology; business processes are predefined, imposed and slightly evolving, and expertise in SOA governance is present. In contrary, public Web services use the Internet, and the latter represents an unsupervised environment but focuses more on end users. Therefore, applying the aforementioned concepts to public Web services may not bring interesting outcome.

Besides, another work is interested in providing reusable data-centric Web services (Saleh et al. 2009) through a formal specification of Web service model. Their model attempts to deal with the interpretation of data that must be marshalled between service consumers and producer systems. It is again an answer to *reusability* of a specific kind of Web service and may not cover all existing categories found among public Web services.

Nonetheless, Bechara and Feuerlicht and Wijayaweera (2007) consider reusable Web services as respecting some general criteria:

- Loose coupling: between services, it refers to minimal dependency among them and between providers and consumers; it means that the service providers shall hide service implementation detail from their consumer by using standard interface (e.g. WSDL, SOAP messages).
- Functional cohesion: a service can contain several operations, and each of them should execute a specific task in order to guarantee the modularity of services.
- Granularity: it is normally hard to achieve. Coarse-grained services return the result after aggregating several response messages. They are then easy to identify. Fine-grained services execute elementary function. They need more breakdown to identify lowest-level elementary actions that they will implement.

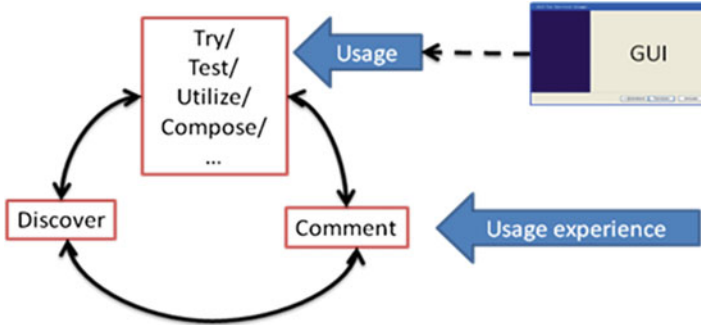


Fig. 50.1 Public web service lifecycle seen by end users

- **Statelessness:** it refers to the fact that during its execution, a service does not wait for any event or callback. There should not be any temporary state during a service execution because it can be very costly to take into account such states.
- **Autonomy:** it refers to autonomous resource usage of a service (e.g. databases, documents).
- **Discoverability:** ability of a service to be identified given a user request that expresses his/her requirement.
- **Composability:** ability of a service to participate in a composition or a mashup scenario given a requirement.

From user point of view, we reduce these *reusability* criteria into usability, discoverability and composability. Usability covers, in effect, loose coupling, functional cohesion, granularity, statelessness and autonomy. This criterion (usability) requires works from service providers to realize, particularly in service design time. Once a service respecting this criterion is published and deployed, it should only be or seen as *runnable* or executable by consumers via testing or try interface provided by service providers. Thus, service usability is designed by service providers and perceived by consumers.

Discoverability and composability, however, can depend both on textual documentation and/or semantic annotation that service providers attach with their services. The mentioned supplementary information can be consulted by service end users but may submerge them into technical difficulty in interpreting or understanding how to use the documented or annotated services. We are facing problems related to common terminology and domain ontology usage between service providers and consumers. Moreover, expertise at both sides is needed in order to work coherently on the service semantic annotations and to construct correct semantic discovery or composition requests.

In the actual work, we choose to tackle the public Web service *reusability* from end-user point of view by encouraging end users to express their service usage experience for the benefit of the others in the community. We initiate our proposal based on the following observation (see Fig. 50.1):

We consider that end users can observe what happens when they use a service and can report what goes through their perceptions to the others and make them taking profit from previous usage experiences to enhance on one hand the *reusability* of Web services and on the other hand provide real situational meta-data on published Web services to their communities. Our approach is oriented to socially provided service semantics as advocated by Averbakh et al. (2009), Jansen et al. (2008), and Leitner et al. (2009). We then have to provide possibilities to end users, allowing them to express their usage experience in a simple way and to exploit those experiences to promote public Web service *reusability*.

3 Expressing Service Usage Experience

We define the process of providing service usage experience as a socialization one. It is based on a simple observatory hypothesis which is “more a service is socialized, more it has been tested or tried”. If a service is repetitively used in various situations, then it is considered as reusable. In this case, functional aspect interests us more than non-functional one. In effect, the latter is often use to compute the selectivity of Web services.

We opt for collaborative socialization process, i.e. users can comment on the services they have been using, and those comments are available to other users. Comments from different users on a service are aggregated to reflect the importance of the commented service. Thus, collaborative tagging (Papathanasiou et al. 2009) is a relevant option for expressing this kind of feedback after using, testing or trying a service.

A service (taggable object) can be tagged by several users with several tags. A tag can be used by several users to tag several services. Therefore, the tagging model can be summarized as a triplet $\langle User, Tag, TaggableObject \rangle$ (see Fig. 50.2). The model illustrated above can be stored in the *folksonomy base* in our supporting system.

Expressing a service usage experience consists then of reporting on it usage situation, what operation has been used, what input is required and what output is produced. Such information can be observed during a test or a try on a service.

According to the structure of WSDL that is normally used to describe Web services, a Web service can be considered as an object that can be parsed into several operations, and each of them is respectively into several pairs of input and output. We allow end users to express such features related to a service they used in their usage experience by focusing particularly on the functional capacity of the service (see Fig. 50.3).

Using tags is both flexible and easy for nonexpert consumers. However, synonymy of terms can generate ambiguity in providing the correct meaning of a used term. Therefore, our tag is associated with a meaning. Each tag has a weight which is its occurrence number, and this weight reflects the importance of the tag related to the tagged service. We impose a simple tagging convention by using “;” as

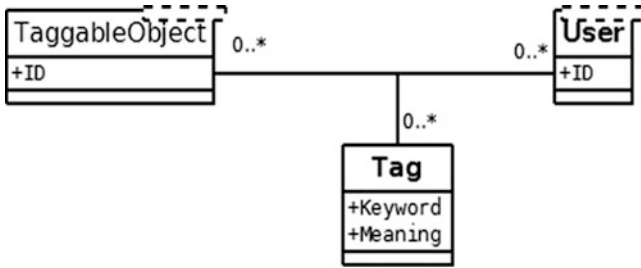


Fig. 50.2 Folksonomic tagging model

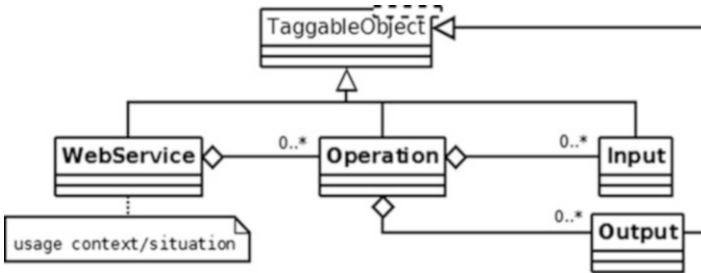


Fig. 50.3 Different taggable item in a Web service

separator, “{ }” as block definition and “_” for multiple term tag (see Fig. 50.4 for an example: *{flight, sense{flight, 0}, 10}* means that the tag “*flight*” is associated to the first meaning found in the term meaning base (*indexed by 0*) and is tagged *10* times on the service).

The next section depicts how we can exploit the user-provided annotation on Web services to promote and enhance the *reusability* of public Web services. We emphasize on the fact that a reusable Web service is most likely usable, discoverable and composable. These criteria should be perceived by end users within public Web service lifecycle in order to encourage them to use and then reuse the published Web services.

4 Public Web Service Reuse Supporting Framework Architecture

The main goal of this supporting framework is twofold: service providers can use it to advertise their published Web services with help from end-user community by making their service more and more usable, discoverable and composable. Service consumers, via this framework, should be able to test, try, use, discover, compose, substitute and annotate published Web services (see Fig. 50.5).

```

WS1:
+ used in: { {{flight, sense<flighth, 0>}, 10};
             {{confirmation, sense<confirmation, 0>}, 5};
             {{booking, sense<booking, 0>}, 2}
           }
+ with an operation performing: {{{send, sense<send, 0>}, 12};
                                 {{sms, sense<sms, 0>}, 20};
                                 {{text_message, sense<text_message, null>}, 8}
                               }
+ by consuming: { {{flight, sense<flight, 0>}, 5};
                  {{number, sense<number, 0>}, 10};
                  {{phoneNumber, sense<phoneNumber, null>}, 3}
                }
+ and producing: { {{sms, sense<sms, 0>}, 10};
                   {{flight, sense<flight, 0>}, 4};
                   {{schedule, sense<schedule, 0>}, 3}
                 }
    
```

Fig. 50.4 Tags associated to WS1

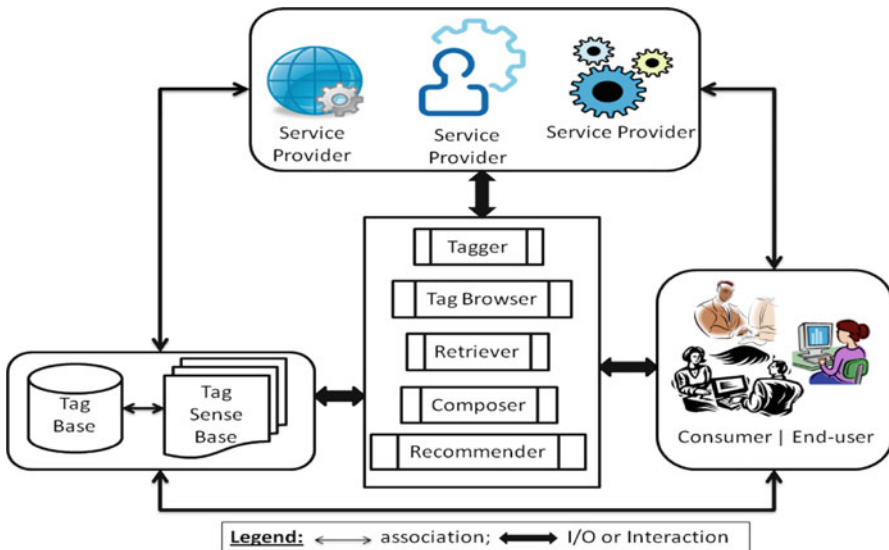


Fig. 50.5 Supporting framework for public Web service reuse

This system is divided into several modules. It also depends on a tag sense base that contains not only a terminological relationship base between tags such as synonym but also for each tag an array of possible meanings associated to that tag. Several existing lightweight knowledge bases or thesaurus can be used to

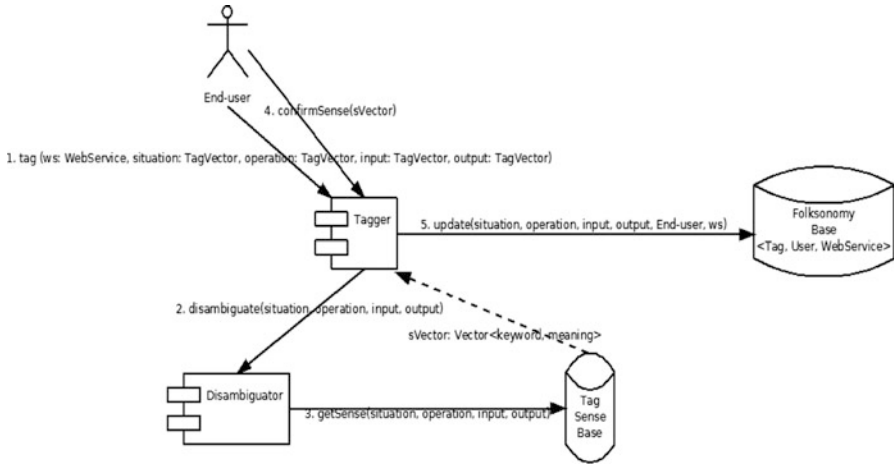


Fig. 50.6 User-system interaction while expressing a service usage experience

implement our tag sense base such as WordNet,² Wikipedia, Common Tag³ or DBpedia.⁴ However, we cannot use exclusively one of them to disambiguate in the tagging process. In effect, on one hand, we search for simple visual sense array proposed to end users while they are entering a tag having multiple senses and, on the other hand, we aim at finding the tag relatedness such as synonymy relation in order to extend the Web service discovery and composition space. Using WordNet with one among the rest of the aforementioned candidates seems very plausible.

Tagger module allows end users to provide comment, feedback or usage experience (tag or annotate) on a Web service. Each given tag is disambiguated thanks to tag sense base, and finally, the triplet $\langle user, tag, annotated\ item \rangle$ is stored in the *folksonomy base* (see Fig. 50.6).

Tag browser module allows end users to visualize tagged Web services and the end users having tagged services given a selected tag to browse. This module gives an insight on what tags have been used and by what users before commenting a service. It can help the tagging user to choose the already used tags to express his/her usage experience (see Fig. 50.7).

Retriever module allows end users to discover tagged Web services based on tags. Looking up Web services this way is more extensive and richer than using simple keyword because in our case, tags are disambiguated and bundled into a structure. We then exploit the lexical and structural information related to the searched services. Moreover, inside the service usage experience expression, usage situation is given, and it can help enhance the quality of discovery process. A discovery request must be formalized as how tags are organized around services.

² <http://wordnet.princeton.edu>

³ <http://commontag.org/Home>

⁴ <http://dbpedia.org/About>

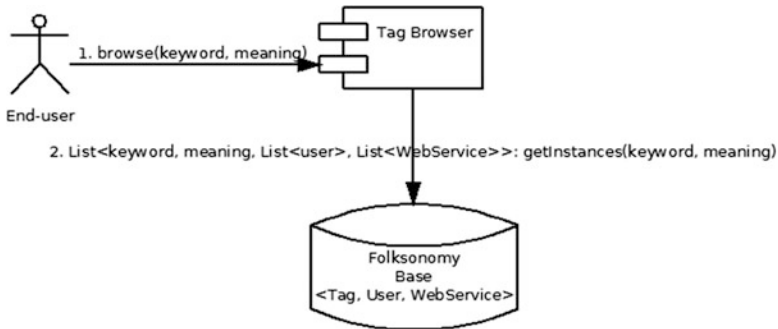


Fig. 50.7 User-system interaction while browsing a tag for services or users

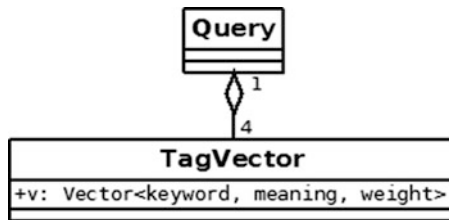


Fig. 50.8 Discovery request model

It is represented by four tag vectors (usage situation, operation, input and output), but each tag item has no weight (see Figs. 50.8 and 50.9). Tag-wise similarity is the core of the discovery algorithm, and we define that two tags are identical if and only if they share the same meaning and the same term. Two tags are similar if they have synonymy relatedness or their meanings reveal a certain degree of textual similarity (e.g. computed by co-occurrence, cosine, Jaccard or dice measures) (Cattuto et al. 2008).

Recommender module can be used as the extension of the three above modules. Its role is to exploit the co-occurrence between tags, determine which tag appears frequently with another, compute closeness of user profile generated from tagging activity in order to propose similar service tagged by user having the similar profile and propose other services tagged by similar tags.

Composer module allows end users to establish a visual composition of a number of services by drag-and-drop actions. After a service is dropped in the composition space, the system has to look up for the most appropriate candidate to pipe from its output by exploiting the exact or similar tag match. This module is served as a concept-proof of likely composability of a chosen service. It is still very hard to achieve real composite services this way due to input, output and constraint on manipulated data type.

```

Find a service:
+ used in: { {flight, sense<flight, 0>;
              {confirmation, sense<confirmation, 0>}
            }
+ with an operation performing: { {send, sense<send, 0>;
                                   {sms, sense<sms, 0>}
                                 }
+ by consuming: { {flight, sense<flight, 0>;
                  {number, sense<number, 0>;
                  {phoneNumber, sense<phoneNumber, null>}
                }
+ and producing: { {sms, sense<sms, 0>;
                   {flight, sense<flight, 0>;
                   {schedule, sense<schedule, 0>}
                 }

```

Fig. 50.9 Textual example of discovery request

5 Conclusion and Future Works

The main contribution of this chapter lies in the identification of service *reusability* features. We then propose a tagging model allowing service consumers to express their usage experience by using tags. We bring in the concepts of using tags to enhance or promote the likely usability, discoverability and composability of services. We propose also some service recommendation mechanisms and the concept of visual service composition.

Then, we provide the architecture of a supporting system that allows end users and service providers to easily enhance and promote public Web service reuse. It can also be a potential candidate to be employed for realizing different tasks such as:

- Web service discovery at design time: service-oriented application engineers can dispose of such system to assist his/her task in finding services that are relevant to the functional requirement of clients. Such system allows engineers to browse different services in the enterprise collection by simple tags and formulate easily his/her request by just following a simple guide. Usage experience of services can be shared among engineer community to improve the reuse of locally available Web services, if those engineers care enough participating in the tagging activities after testing or using the services.
- User-assisted Web service composition: 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 service. The need formulation is easy, and our tagging model favours the annotations on input/output of Web services which are habitually useful for the composition process.
- User-oriented and semi-automatic Web services clustering or classification: with the increasing number of tags on functionality of services provided by users, the

expression of actual usage of services can be deduced. By employing efficient text or data mining algorithms, the clustering or classification of such annotated service collection is more than feasible and can bring interesting outcome.

- User-centric Web service recommendation: collaborative tagging provides meta-data in form of tags which can be exploited in the recommendation process based on collaborative or user profile-based filtering.

However, dealing with tags provokes some issues also such as tag bombing or tag spamming. We do not tackle such issues in the scope of this chapter.

As the perspective of our proposal, technical issues need to be studied in order to implement an appropriate system based on our aforementioned architecture. Several elements need to be considered such as the complexity of tag disambiguation process, the quality and scalability of service discovery and composition based on tags. Service recommendation process should be studied in more detail in order to exploit the minimum amount of data for the acceptable results.

To sum up, we attempt to promote the likelihood of public Web service reuse by socializing those services. Service providers can take benefit from such approach without heavy additional work. End users are however encouraged to express their service usage experience in a collective and collaborative environment in order to promote service *reusability*.

References

- AbuJarour M, Naumann F, Craculeac M (2010) Collecting, annotating, and classifying public web services. In: Proceedings of the 2010 international conference on the move to meaningful internet systems, volume part I. Springer, Berlin/Heidelberg, pp 256–272
- Averbakh A, Krause D, Skoutas D (2009) Exploiting user feedback to improve semantic web service discovery. The Semantic Web – ISWC 2009, pp 33–48
- Bechara G . What is a reusable service? <http://www.oracle.com/technetwork/articles/bechara-reusable-service-087796.html>
- Cattuto C, Benz D, Hotho A, Stumme G (2008) Semantic grounding of tag relatedness in social bookmarking systems. In: Sheth A, Staab S, Dean M, Paolucci M, Maynard D, Finin T, Thirunarayan K (eds) The semantic web – ISWC 2008. Springer, Berlin/Heidelberg, pp 615–631
- Chu W, Qian D (2009) Design web services: towards service reuse at the design level. J Comput 4:193–200. Academy Publisher
- Dan A, Johnson RD, Carrato T (2008) SOA service reuse by design. In: Proceedings of the 2nd international workshop on systems development in SOA environments. ACM, New York, pp 25–28
- El-Goarany K, Saleh I, Kulczycki G (2008) The social service network – web 2.0 can make semantic web services happen. In: 10th IEEE conference on E-commerce technology and the Fifth IEEE conference on enterprise computing, E-commerce and E-services, 2008, pp 419–423
- Fan J, Kambhampati S (2005) A snapshot of public web services. SIGMOD Rec 34:24–32
- Feuerlicht G, Wijayaweera A (2007) Determinants of service reusability. In: Proceeding of the 2007 conference on new trends in software methodologies, tools and techniques: proceedings of the sixth SoMeT 2007. IOS Press, Amsterdam, pp 467–474

- Hageman S, Letz C, Vossen G. Web service discovery – reality check 2.0.
- Jansen S, Brinkkemper S, Hunink I, Demir C (2008) Pragmatic and opportunistic reuse in innovative start-up companies. *IEEE Softw* 25:42–49
- Leitner P, Michlmayr A, Rosenberg F, Dustdar S (2009) Selecting web services based on past user experiences. In: 2009 I.E. Asia-Pacific services computing conference, Biopolis, Singapore, 7 décembre
- Li Y, Liu Y, Zhang L, Li G, Xie B, Sun J (2007) An exploratory study of web services on the internet. In: IEEE international conference on web services, 2007. ICWS 2007. IEEE Computer Society, Los Alamitos, pp 380–387
- Loutas N, Peristeras V, Tarabanis K (2009) Rethinking the semantic annotation of services. In: Proceedings of the 2009 international conference on service-oriented computing, pp 540–549
- Papathanasiou MK, Loutas N, Peristeras V, Tarampanis K (2009) Combining service models, semantic and web 2.0 technologies to create a rich citizen experience. In: Proceedings of the 2nd world summit on the knowledge society: visioning and engineering the knowledge society. A web science perspective. Springer, Chania, Crete, Greece, pp 296–305
- Petrie C (2009) Practical web services. *IEEE Internet Comput* 13:93–96
- Saleh I, Kulczycki G, Blake MB (2009) A reusable model for data-centric web services. In: Proceedings of the 11th international conference on software reuse: formal foundations of reuse and domain engineering. Springer, Berlin/Heidelberg, pp 288–297
- Tanaka M, Murakami Y, Ishida T (2008) Towards service supervision for public web services. IEEE Asia-Pacific services computing conference, 2008. APSCC 2008, pp 536–541
- Wu C, Chang E (2007) 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, pp 321–328