Context, Quality and Relevance: Dependencies and Impacts on RESTful Web Services Design*

Hong-Linh Truong[1], Schahram Dustdar[1], Andrea Maurino[2], and Marco Comerio[2]

[1] Distributed Systems Group, Vienna University of Technology, Austria
{truong,dustdar}@infosys.tuwien.ac.at
[2] Department of Informatics, Systems and Communication
University of Milano - Bicocca, Italy
{maurino,comerio}@disco.unimib.it

Abstract. While several techniques have been introduced for specifying and acquiring context and quality information associated with Web services, they consider such information representing the whole Web services. However, accessing to context and quality of data resources provided by Web services is crucial. This is particularly relevant for data-intensive Web services of which the context and quality of data resources will strongly impact on the service development and composition. In this paper we contribute an analysis of relationships among context, quality, and relevance, as well as their impact on the design and composition of Web services, in particular at the data resource level. Then, we propose several techniques to incorporate context and quality descriptions into REST APIs and RESTful services publishing. By implementing these features, RESTful Web services could allow the consumer to specify and query context and quality information associated with services and data resources, thus fostering the provision of high relevant data resources.

1 Introduction

Web services (WS) have fostered the access of data resources in the Internet scale, e.g. several datasets are available in the UN DATA API project[1] or Infochimps[2]. When discovering, composing and executing Web services, we consider them and their provided data resources. However, currently non-functional descriptions of Web services mainly represent the service as a whole and they provide only marginal information at the data resource level. Several models and techniques [1,2] have been proposed to extend standard descriptions of a Web service with information regarding context (e.g., conditions for the service usage) and quality of service (e.g., response time, availability). Vice versa, there is not the same support for the specification of context and quality of data of data resources that can be accessed through the Web services, such as (i) under which situation the data can be used and (ii) the accuracy of the data. In particular, this problem is relevant for data-intensive RESTful Web services which rely on simple

* The research in this paper is partially funded by the EU under the FP7 Commius project and by the SAS Institute srl (Grant Carlo Grandi).

[1] http://www.undata-api.org/

[2] http://api.infochimps.com/

F. Daniel and F.M. Facca (Eds.): ICWE 2010 Workshops, LNCS 6385, pp. 347–359, 2010.

principle and operation patterns and present several advantages over SOAP-based Web services [3]. RESTful Web services represent a practical way to access data resources[3] on the Web but currently lack techniques for acquiring quality and context information associated with data resources and services, thus it is difficult to improve the relevance of the outcome of RESTful Web services.

The outcome of a service is considered relevant when the provided results match the consumer's purpose. Naturally, the concept of *relevance* is dependent on the consumer's need. To provide relevant results to the consumer, a service must be able to handle (i) context and quality information associated with the consumer's need and (ii) context and quality information associated with the provided resources. Based on that, any service composition should utilize context and quality descriptions from both sides - the provider and the consumer - in order to support the concept of relevance. This principle is not new, as researchers have investigated several ways to improve the relevance of output of different systems, such as information retrieval systems [4,5]. However, there is a lack in understanding the dependency among context, quality and relevance in data resource-oriented Web services. Furthermore, often the design of Web services lacks guidelines for supporting context and quality aspects at the data resource level. This problem strongly hinders the development of algorithms and techniques to improve the relevance of results provided by data resource-oriented services.

To tackle the above-mentioned issue, in this paper we focus on data-intensive services with which the relevance of the service output is judged mainly, besides the functionality of the data, on the context and quality in which the data can be utilized. We contribute an analysis of relationships among context, quality, and relevance, as well as their impact on the design and composition of Web services. Then, since the REST model is increasingly used for data-intensive services, we propose several techniques to incorporate context and quality descriptions into REST APIs and RESTful services publishing to allow the consumer to specify and query context and quality information associated with services and resources, thus fostering the evaluation of the relevance degrees of RESTful services and data resources.

The rest of this paper is organized as follows. Section 2 presents our motivating scenario. We present the role of context, quality, and relevance in Section 3. Current supports of RESTful services with respect to the context, quality, and relevance are discussed in Section 4. Techniques to enhance the support of quality and context in RESTful designs are presented in Section 5. Related work is discussed in Section 6, followed by a conclusion and future work in Section 7.

2 Motivating Scenario

To examine the importance of context and quality information related to Web services and data resources during service design, composition and execution, let us consider a simple composite service that supports the search of news and images. The composite service includes three RESTful Web services named `Yahoo! Boss News`

[3] Since we focus on services providing data, in this paper the two terms "resource" and "data resource" are used interchangeably.

`Search`[4], `Google News Search`[5], and `Flickr`[6]. Our composition is written using Yahoo! Pipes[7] which invokes these Web services using their REST APIs. Given a query of key words from the user, while the composition invokes services which return many data resources (e.g., figures and web news), many of them might not be relevant to the current user's context and expected quality.

Let us imagine that the user of the composite service is a doctor and she would like to perform a health research in Haiti due to the recent earthquake in January 2010[8]. In her research, she would like to find recent news and high-qualified images, but free-of charge. By entering `"Haiti"` into the composite service, the composite service is able to find news and images. Let us consider the first case in which the composite service does not understand the context of the search or does not understand how to specify context and quality parameters in service APIs. The corresponding composition is shown in Figure 1. In the second case, let us assume that the composite service understands context of the user or the composite service is able to obtain the user's context and quality requirements (this can be achieved based on specific ways of how the composition execution platform interacts with the user). Furthermore, the composite service knows how to specify context- and quality-related parameters in REST APIs of its composed services. Using the context and quality description expected by the user, the composite service can utilize these descriptions in the invocation of Web services and the processing of results to the user. Table 1 presents possible mappings from the user's requirements to parameters of REST APIs of corresponding services in the composition and Figures 2 shows the composition that utilizes appropriate context and quality parameters. We have observed that, in the first case, several results are not relevant to the user's request. Between two invocations, the results from `Yahoo!Boss` are similar as the quality parameter used does not affect the results of the five most relevant news. However, in the second invocation, results from `Google News Search` have a higher *irrelevance* as news are not related to Haiti and images from `Flickr` are more *relevant* based on understanding the *context* of the user and on the utilization of *quality* description of data resources provided by services.

The scenario shows the importance of understanding and utilizing context and quality information in order to provide relevant results to the consumer. However, this also presents many existing issues in current RESTful service design. First, how can the composition developer recognize context and quality description associated with services? For example, does `Flickr` service support data quality metrics or not? If not, maybe `Picasa` service[9] should be used. Second, how context and quality parameters can be mapped and passed to a service via REST APIs (e.g., when asking services to provide high accuracy images)? Third, how to obtain context and quality description associated with services and data resources so that further activities can be made

[4] `http://developer.yahoo.com/search/boss/boss_guide/`
[5] `http://code.google.com/apis/ajaxsearch/documentation/reference.html#_fonje_news`
[6] `http://www.flickr.com/services/api/`
[7] `http://pipes.yahoo.com`
[8] `http://en.wikipedia.org/wiki/2010_Haiti_earthquake`
[9] `http://picasaweb.google.com/`

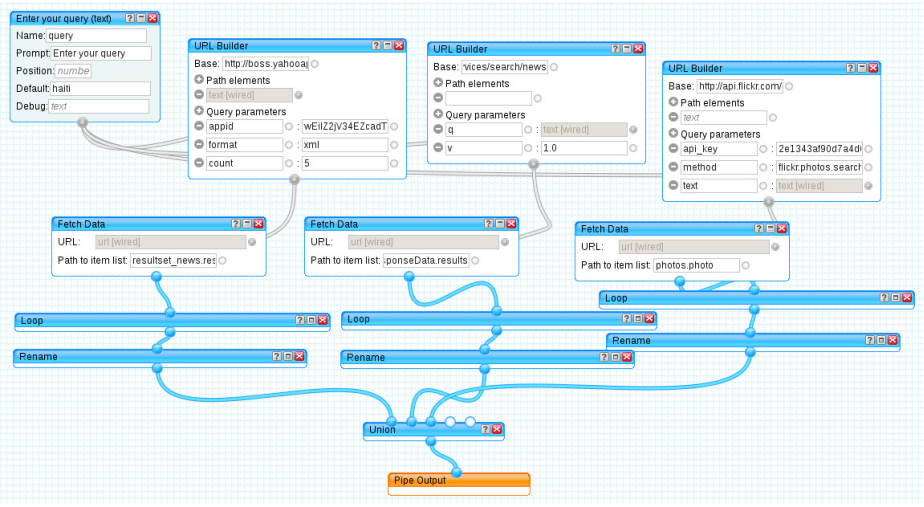

Fig. 1. A composite service for searching news and images

Table 1. Possible mappings of context and quality requirements to REST APIs. The mapping is not necessary suited to the provider's definition but it is based on the view of a service composition developer.

Type	Yahoo!Boss News Search	Google News Search	Flickr
Context	`age=2w` (the news in the last two weeks)	`contacts=all` (search in user's contacts);`topic=m` (search on health news)	`licenses=1,2,3` (types of licenses are Attribution-NonCommercial-NoDerivs, Attribution-NonCommercial, and Attribution-NonCommercial-ShareAlike Licenses); `tag=heath,medical` (for medical topic), `min_taken_date=2010-01-10` (for recent images)
Quality	`age=2w` (the timeliness of the news)	`scoring` (the higher the score is, the more relevant the news is)	`accuracy=3` (images with the country level), option `b` for selecting high resolution images

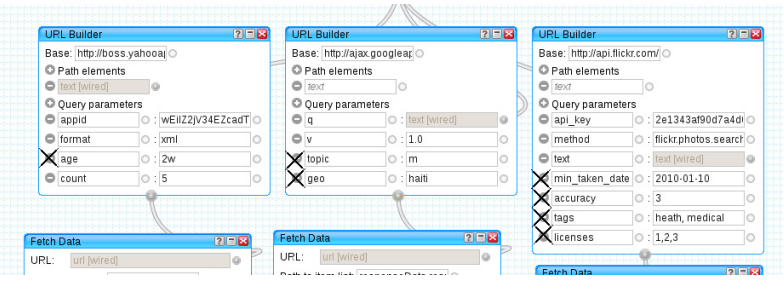

Fig. 2. Adding context and quality parameters (marked by `X`) during the service composition

in the service composition (e.g., filtering relevant images and news records based on their quality description)? When solutions for such questions are provided, the service composer can adapt and improve the quality of compositions substantially. However, this cannot be achieved without understanding the impact of having quality and context information at the resource level on that of services and composite services.

The two examples show that missing context and quality information leads to irrelevant results. Context and quality are not only associated with consumers but also with services and data resources. Being able to obtain context and quality information at the resource level will help to provide highly relevant results to consumers. However, so far, little attention has been spent to support this issue.

3 Context, Quality, and Relevance Dependencies

In principle, a composite service utilizes Web services that provide mechanisms to retrieve and manipulate resources. A consumer utilizes a composite service in order to retrieve the (most expected relevant) data resources. To be able to provide relevant data resources to the consumer, the composite service, Web services and data resources may implement interfaces to support the processing, publishing and discovery of possible context and quality information. In the execution model of the composite services, Web services and data resources, two flows of context and quality information exist. First, context and quality information can be required and defined based on the $consumer \rightarrow composite\ service \rightarrow Web\ service \rightarrow data\ resource$ flow. Second, such information can be aggregated and changed based on the $data\ resource \rightarrow Web\ service \rightarrow composite\ service \rightarrow consumer$ flow. Along these flows, several operations can be applied to context and quality information in order to support context- and quality-aware services and to provide high relevant results. We will concentrate on the second flow in this paper.

Figure 3 shows that context and quality are associated with composite services, Web services and data resources but they are originated from two different sources: the provider/integrator and the consumer. The consumer's context and quality descriptions define what an individual consumer need. The context and quality descriptions from

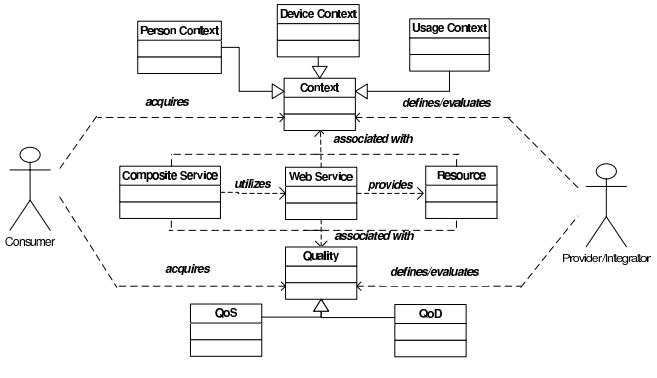

Fig. 3. Context and quality associated with composite services, Web services, and data resources

Table 2. The impact of the lack of context and quality information on the relevance of the (composite) service's output

Type	Context	Quality	Impact on Relevance
Resource	not specified	not specified	it is difficult to select relevant data resources and to provide general description of context and quality for the service managing the resource. The service has to build its own quality and context description. This could be very difficult if the service developer is not the resource provider (e.g., the developer aggregates data resources in the Web).
Resource	specified	not specified	it is not clear if the resource will meet the consumer quality description. The service has to use its own knowledge to determine the quality of the resource.
Resource	not specified	specified	it will not be clear whether the data resource can be used. The service can only rely on its own knowledge in order to determine if a resource is suitable for a particular context.
Web Service	not specified	not specified	it is not sure if the Web service can be used. Even if the Web service can be used, it is not sure if the resource provided can be used. The composite service has to implement its own service selection mechanisms.
Web Service	specified	not specified	the Web service can be used but the composite service is not sure about the quality of the service and the resource. The composite service has to determine the quality of the resources at its side using its own knowledge.
Web Service	not specified	specified	it is not clear if the Web service is suitable for the context. The composite service has to determine the context of the service by its own knowledge.

the provider describe how services and data resources fit to generic requirements. As shown in Figure 3 many types of context and quality can be specified. The context can be specialized in *person context* (i.e., the context associated with consumer/provider), *device context* (i.e., the context of the device used to access the composite service, Web service or data resource) and *usage context* (i.e., the context in which the composite service, Web service or resource is supposed to be used). The quality can be specialized in *quality of data (QoD)* (e.g., the accuracy of data resources) and *quality of service (QoS)* (e.g., response time of the Web service).

Although several techniques have been introduced for specifying and acquiring context and quality information for services, they focus on the service level. Therefore, context and quality associated with data resources are also crucial in order to provide high relevant results. Table 2 describes how the lack of quality and context information at the resource level (and relevant to service level) impacts on the implementation of composite services able to provide high relevant results. Overall, this lack of information forces the service or the composite service to develop several mechanisms to compensate the missing information and these mechanisms might rely on only the knowledge of the service composition developer. This requires us to incorporate and develop models and APIs for handling context and quality information. The description, specification, and evaluation of context and quality information and the utilization of such information for improving the relevance of the service output require several research activities.

4 Quality and Context Support in Current RESTful WS Design

Having a detailed analysis of the dependencies among context, quality, and relevance, we examine how such information is coupled and supported in the current RESTful service design. The REST architectural model assumes that a resource can be created,

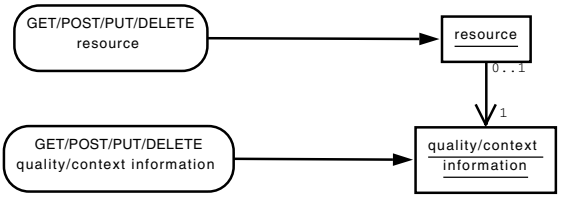

Fig. 4. Separate handling for resources and their context/quality information

updated, modified, and deleted by using four operations named `GET, POST, PUT`, and `DELETE`. Such operations are not designed to handle quality and context descriptions associated with resources. Of course, in principle, one can use these operations to handle context and quality associated with a data resource. However, so far we are not aware of this in the design of RESTful services. One approach is to consider the quality and context information as a new resource and access them separate from the resource they characterize (see Figure 4). However, in this case, the consumer will have difficulties to understand and manage the relationship between the resource and its quality/context description via separate resource paths in REST APIs. Therefore, to support the consumer to specify and obtain context and quality description of resources, similar operations on a resource should be used for handling the resource (such as `GET` the resource) and for handling context/quality information (such as `GET` the data quality).

Currently, at the service level, the context and quality information associated with RESTful services and parameters for specifying context and quality requirements cannot easily be distinguished from the REST APIs, if not impossible. In fact, many RESTful services do not provide such information. In RESTful service descriptions, such as based on WADL[10] or ATOM[11], there is no context and quality information. At the REST APIs level, there is no convention for specifying and obtaining context and quality information. Therefore, the RESTful service design should provide context and quality description exchange protocols among the interactions of service compositions, Web services and resources. In the following section, we present our approach in detail.

5 Enhancing Context and Quality Support in RESTful WS Design

In this section, we propose some extensions to the design and implementation of RESTful services in order to support quality and context aspects. We present our guidelines for this purpose by means of an experimental service which is a RESTful Web service to provide data from the Google Flu Trend[12]. In our experimental service, named `googleFluTrend`, the list of countries is considered as a data resource and each country is also considered as a data resource[13].

[10] `https://wadl.dev.java.net/`
[11] `http://www.ietf.org/rfc/rfc5023.txt`
[12] `http://www.google.org/flutrends/`
[13] Due to the space limit, we provide detailed supporting materials at `http://www.infosys.tuwien.ac.at/prototyp/SOD1/quacore`

5.1 Representations for Context and Quality

Several works have proposed context and quality descriptions defined using, for example, XML, RDF and OWL [6,7,8]. Therefore, they can be reused to define representations for RESTful service. Since XML and JSON are the most popular way of describing the requests, responses and descriptions of RESTful services, we use them to represent the context and quality information.

Our quality and context representations are based on our work in context and concerns for data as a service [9]. Listings 1 presents an excerpt of a custom quality description in JSON (readers could refer to our supporting materials for examples of custom quality and context models). Given the representations of context and quality for RESTful services, we propose the following techniques for coupling context and quality information with RESTful Web services.

```
{
 "crq.qod": {
   "crq.uptodateness": "up to dateness",
   "crq.objectivity":"objectivity",
   "crq.freeoferror":"free of error",
   "crq.consistency":"consistency",
   "crq.dataelementcompleteness":"data element completeness",
   "crq.datasetcompleteness":"data set completeness",
   "crq.domainspecificqod":"URI specifying domain specific info"
 },
 "crq.qos":
 {
   "crq.responsetime":"response time",
   "crq.latency": "latency",
   "crq.capability":"capability",
   "crq.reliability":"reliability",
   "crq.availability":"availability"
 }
}
```

Listing 1. Simplified JSON-based quality representation

5.2 Coupling Context and Quality with RESTful Services

Given their representations, context and quality information need to be associated with resources and services and to be published so that the service composition can utilize the information. This association is performed by the service provider and the context and quality information reflect the conformity of the services and resources provided by the services. Obviously, such information can be published using existing approaches in Web service information management, e.g., using Web service registries. However, these approaches have not been designed for publishing information characterizing RESTful resources and services. To follow the principle of and the widely-accepted resource description for RESTful Web services, we illustrate techniques to publish context and quality information based on WADL.

WADL allows us to specify the following main elements: `application`, `resource`, `method`, `request`, `response`, and `representation`. Context and quality information can be associated with all these elements with the exception of

representation. From the service provider view, WADL elements can be associated with context and quality description for the following purposes: (i) publishing information for service discovery: this is related to application and resource, (ii) allowing service consumers to specifying inputs of context and quality in REST APIs: this applies to request and method, and (iii) allowing service consumers to obtain context and quality information associated with resources by using REST APIs: this applies to the response and method.

Publishing quality and context information for service discovery: In order to publish the context and quality description associated with a service, we need to provide two types of information (i.e., the schemas and the information according to the schemas) and to associate them with RESTful Web services. Because the context and quality descriptions are considered as extra documents about RESTful Web services, we should map these schemas and descriptions by using optional elements in WADL. To utilize existing constructs of WADL, we propose the following guidelines. The mapping of schemas of context and quality can be performed by using the grammars element of WADL. This element allows to include external schemas using a sub-element include. The representation element can be used to describe published, static quality and context descriptions. Furthermore, to distinguish different schemas and types of information, we can use the doc element. Table 3 describes how to associate quality and context information with the WADL of the service.

Table 3. Describing context and quality information in WADL

Element	Usage	Example
grammars/include	specify context and quality schemas	<include href="crq-quality.xsd">
grammars/include/doc	specify the type of schema. We propose to use the title to describe the name of schema	<doc title="Quality">
representation	specify the published, static quality and context information in service description and specify the representation of context and quality information associated with REST APIs (e.g., GET and POST)	<representation id="QoD" element="crqQuality:crd.qod" mediaType="application/json">
representation/doc	specify the content of static, published quality or context information.	<doc title="QoDDescription">... </doc>

Based on the above-mentioned descriptions, several activities can be performed. For example, if a service consumer wants to search services based on quality and context descriptions, the consumer can utilize the information specified in the element representation/doc by filtering relevant representation elements based on the title element. Consumer-side code generation tools can utilize the representation, request and response elements to generate codes for handling quality and context information.

Example: A WADL description is available at http://www.infosys.tuwien.ac.at/prototyp/SOD1/quacore/examples/GoogleFluTrend-v0.2.wadl for our experimental service. The wadl2java tool[14] can be used to generate

[14] https://wadl.dev.java.net/

functions `getAsQoD(String crqQod, String crqContext)`, and `getAsService Context(String crqQod, String crqContext)` to obtain context and quality information.

Specifying Context and Quality Parameters in REST APIs: The specification of context and quality descriptions in REST APIs can be done by using query parameters, which can be built based on the name described in the `request` of the WADL. By utilizing query parameters the form of `crq.metricName=value`, where `crq.metricName` is the name of context and quality metrics specified in context and quality representations, one can indicate context and quality aspects in REST APIs.

Example: To select resources with the minimum accuracy 0.5 in Europe, we can use the request `GET/resource?crq.accuracy="0.5"&crq.location=''Europe''`. Given the input request of context and quality information, the service can utilize these information to select the right resources. Optionally, the response can also include quality and context related information together with the requested resource.

Obtaining Context and Data Quality in REST APIs: Context and data quality descriptions should also be obtained for services and resources without obtaining the services and resources. To this end, we propose to use query parameters. For this purpose, context and quality parameters are specified but without any values. By following this convention, we can assume that the consumer requires only context and quality information. For example, a request like `GET/resource?crq.qod` would return only the quality of data of the requested resource. Using this way, the service consumer can query only the context and quality information before deciding which resources it should access. An advantage is that resources will not be accessed if their quality is not guaranteed. However, in cases of context- and quality-guaranteed resources, two invocations are needed to retrieve the resources.

Example: while the resource containing all countries can be obtained by using `curl http://.../resources/googleFluTrends`, to obtain the quality of this resource the parameter `crq.qod` without any value can be used as follows:

```
$curl http://localhost:8080/restfuldesign/resources/googleFluTrends?crq.qod
{"crq.qod": {
     "crq.datasetcompleteness": 0.10256410256410256,
     "crq.consistency": 1
}}
```

A request of `GET/googleFluTrends?crq.dataelementcompleteness=0.9` will return only resources that have the minimum completeness 0.9. For example, the data resource `Austria` is returned as its quality of data is:

```
{"crq.qod": {
     "crq.dataelementcompleteness": 0.9,
     "crq.consistency": 1
}}
```

6 Related Work

The majority of related work with respect to the role of context, quality, and relevance is in the focus of the information retrieval. Knight and Burn have proposed the relevance as a data/information quality metric and a contextual metric [10]. Batini and colleagues have also presented the fact that the relevance is also used as a data quality metric for data provided by databases [7]. However, these discussions are limited to database aspects. The relevance term defined in [7,10] can be used to indicate on the relevance of the resource from the provider view. Lachica and colleagues have proposed a framework that uses quality and relevance to rank information in information retrieval systems [11]. In their work, relevance is context dependent and is characterized by four types of context information. Thus, in some senses, they also presented the relationship among context, relevance and quality. However, their work is not focused on Web services and their relevance term is independent on quality information.

In Web services, user context and quality of service have been long considered as valuable source of information for supporting Web service design, discovery and composition [12,13]. Most related works using quality and context information in service-oriented computing can be divided into three classes: (i) approaches to enhanced Web service design, such as different ways to add non-functional parameters to Web services in [14]; (ii) techniques to increase the relevance of the result of Web service discovery and selection, such as the discovery of resources based on WS-Policy specifications using an external middleware in [15] and the NFP-based hybrid approach to Web service ranking in [16]; (iii) approaches to improve the relevance of information offered by the services using context information, such as users and their interaction with a service [12], user experiences [17], and context models for personalized Web services [13].

However, existing works in the above-mentioned classes mainly deal with QoS and context information at the whole service level and they do not cover the quality of the information of resources offered by the services. Furthermore, these works mainly support the service/resource discovery. Different from them, we focus on mechanisms to inquire combined context and quality metrics associated with resources. In [18], the quality of mashups and how to evaluate it are discussed. We believe that our approach is a complement work as we provide mechanisms for specifying and accessing quality of data associated with resources. Such mechanisms would simplify the retrieval of quality metrics of resources in order to evaluate data quality metrics for mashups.

7 Conclusion and Future Work

In this paper, we have analyzed the importance of context and quality support in service design, discovery, composition and execution, with a focus on the data resource level. We have discussed the impact of the lack of context and quality information on the guarantee of highly relevant response to the consumer. Although our study is generic, to prove our concepts, we have examined limitations of current RESTful Web service design w.r.t. context and quality information management. To overcome these limitations, we propose several steps in RESTful service design to allow data resources, Web services and composite services to specify and obtain context and quality information.

Our guidelines focus only on mechanisms for inquiring context and quality metrics associated with resources and for requesting and retrieving these resources using these associated metrics. Thus, semantic mapping of context and quality parameters within an individual (composite) service or consumer, is not addressed. The way to specify context and quality descriptions in REST APIs is just one method that will be compared with other ways, such as using HTTP headers.

References

1. Kopecky, J., Vitvar, T., Bournez, C., Farrell, J.: Sawsdl: Semantic annotations for wsdl and xml schema. IEEE Internet Computing 11(6), 60–67 (2007)
2. Patil, A.A., Oundhakar, S.A., Sheth, A.P., Verma, K.: Meteor-s web service annotation framework. In: WWW 2004: Proceedings of the 13th international conference on World Wide Web, pp. 553–562. ACM, New York (2004)
3. Pautasso, C., Zimmermann, O., Leymann, F.: Restful web services vs. "big" web services: making the right architectural decision. In: WWW 2008: Proceeding of the 17th international conference on World Wide Web, pp. 805–814. ACM, New York (2008)
4. Maglaughlin, K., Sonnenwald, D.: User perspectives on relevance criteria: a comparison among relevant, partially relevant, and not-relevant judgements. J. Am. Soc. Inf. Sci. Technol. 53(5), 327–342 (2002)
5. Greisdorf, H.: Relevance: An interdisciplinary and information science perspective. Informing Science 3, 67–71 (2000)
6. Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F.A., Tanca, L.: A data-oriented survey of context models. SIGMOD Record 36(4), 19–26 (2007)
7. Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement. ACM Comput. Surv. 41(3) (2009)
8. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing (2009) (article in press)
9. Truong, H.L., Dustdar, S.: On Analyzing and Specifying Concerns for Data as a Service. In: Proceedings of The 4nd IEEE Asia-Pacific Services Computing Conference, APSCC 2009, Singapore, December 7-11. IEEE, Los Alamitos (2009)
10. Knight, S.A., Burn, J.: Developing a framework for assessing information quality on the world wide web. Informing Science 8, 159–172 (2005)
11. Lachica, R., Karabeg, D., Rudan, S.: Quality, relevance and importance in information retrieval with fuzzy semantic networks. In: Proc. of the fourth international conference on Topic Maps Research and Applications, TMRA (2008)
12. Pernici, B. (ed.): Mobile Information Systems: Infrastructure and Design for Adaptivity and Flexibility. Springer, New York (2006)
13. Maamar, Z., Mostefaoui, S.K., Mahmoud, Q.H.: Context for personalized web services. In: Proc. of the 38th Annual Hawaii International Conference on System Sciences (HICSS), Washington, DC, USA. IEEE Computer Society, Los Alamitos (2005)
14. Ortiz, G., Núñez, J.H., Clemente, P.J.: How to deal with non-functional properties in web service development. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 98–103. Springer, Heidelberg (2005)
15. Mietzner, R., van Lessen, T., Wiese, A., Wieland, M., Karastoyanova, D., Leymann, F.: Virtualizing services and resources with probus: The ws-policy-aware service and resource bus. In: IEEE International Conference on Web Services, ICWS 2009, pp. 617–624 (2009)

16. Comerio, M., De Paoli, F., Palmonari, M.: Effective and flexible nfp-based ranking of web services. In: Proc. of ICSOC/ServiceWave 2009, Stockholm, Sweden, pp. 546–560 (2009)
17. Kokash, N., Birukou, A., D'Andrea, V.: Web service discovery based on past user experience. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 95–107. Springer, Heidelberg (2007)
18. Cappiello, C., Daniel, F., Matera, M.: A quality model for mashup components. In: Gaedke, M., Grossniklaus, M., Díaz, O. (eds.) ICWE 2009. LNCS, vol. 5648, pp. 236–250. Springer, Heidelberg (2009)