Language Service Design Based on User-Centered QoS

Donghui Lin, Toru Ishida and Yohei Murakami

Abstract To design complicated language services for intercultural collaboration, it is important to consider how to compose different atomic language services so as to satisfy the requirements raised by the users. In the research area of services computing, many approaches to QoS-aware service composition have been proposed, where the assumption is that the QoS of a composite service can be calculated by aggregating the individual QoS values of atomic services. However, the QoS of a composite service is difficult to predict due to the QoS uncertainty of services in the real world. In this chapter, we address the above concerns by proposing a service design approach based on user-centered QoS and then confirming it in a case study of language service design for multi-language communication. To achieve this goal, we first introduce a QoS model to address the issues of user requirements and QoS uncertainty. We then propose a service design process that consists of composite service generation, participatory simulation, QoS evaluation and QoS data update. Finally, we introduce a field study of multi-language service design for agricultural knowledge communication to elucidate the benefits of our proposal approach.

Keywords Language service design · User-centered approach · Multi-language communication

D. Lin (✉) · T. Ishida
Department of Social Informatics, Kyoto University,
Kyoto 606-8501, Japan
e-mail: lindh@i.kyoto-u.ac.jp

T. Ishida
e-mail: ishida@i.kyoto-u.ac.jp

Y. Murakami
Unit of Design, Kyoto University,
91 Chudoji Awata-cho, Kyoto 600-8815, Japan
e-mail: yohei@i.kyoto-u.ac.jp

Y. Murakami et al. (eds.), *Services Computing for Language Resources*,
Cognitive Technologies, https://doi.org/10.1007/978-981-10-7793-7_8

1 Introduction

With the development of services computing technologies, service-oriented environments have become more and more popular as they enable users to create, share and combine Web services for various purposes. The Language Grid [11, 12], a typical example of a service-oriented environment for language services, allows multi-language communication services by combining various atomic language services including machine translation services, dictionary services, and so on. Language service design can be regarded as a typical problem of QoS-aware service composition, where the set of atomic services that generate maximum overall QoS value while satisfying some constraints is selected as the optimized solution [1, 7, 31]. In such problems, there is always the assumption that composite services are given beforehand, which might not be reasonable in the real world. Therefore, we focus on designing composite services for real world use, rather than assuming the provision of a composite service for which we then select atomic services.

To design composite services in the real world, we need to deal with the two issues when applying traditional QoS-aware service composition approaches. First, service performance may fluctuate due to dynamic change of service environments in the real world [20] so QoS is inherently uncertain [30], which makes it difficult to design composite service based on historical QoS data. This is especially true if human services are combined with Web services as occurs often in the real world [14, 15, 18], which makes the problem of QoS uncertainty even more complicated. Second, when there are multiple QoS attributes for evaluating services, it is difficult to maximize all the QoS attributes because there might always be counteractive relations between them [2]. For example, improving the translation quality in multi-language communication service might trigger an unacceptable increase in cost. Therefore, it is necessary to design composite service based on user requirements.

As shown in Fig. 1, there are two QoS attributes (Attribute 1 and Attribute 2) that counter each other. Users have their own requirements on the attributes and there are several candidate composite services. However, only an approximate QoS range can be obtained from the QoS data for each composite service due to QoS fluctuations. It might not be appropriate to utilize average QoS values when designing the composite services due to uncertainty in the distribution of the QoS values. This is exactly what happens when designing a multi-language communication service in the real world. Therefore, we try to design composite services while being aware of the above issues. A previous study noted that the user-centered approach is important in creating interactive systems [13]. In this chapter, we aim at a service design approach based on user-centered QoS, where users' satisfaction is evaluated throughout the whole design process. To achieve this goal, we first describe a QoS model for service design that takes into consideration the user requirements by extending existing QoS models for service composition. Then, we propose a participatory service design process that consists of composite service generation, participatory simulation, QoS evaluation and QoS data update. Finally, we use a field study of multi-language communication service design to validate our proposed design approach.

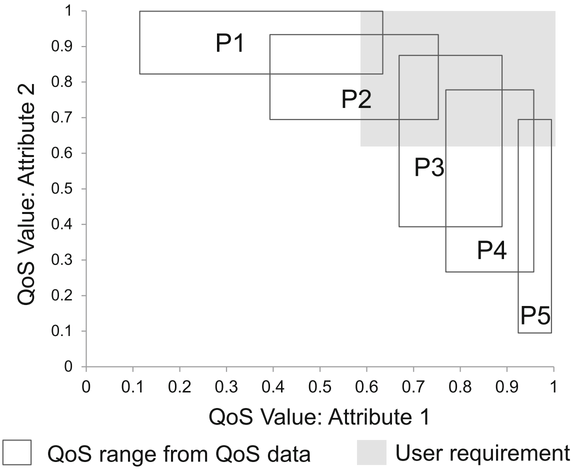

Fig. 1 Counteractive relations between two QoS attributes (P1-P5 are alternative composite services)

The rest of this chapter is organized as follows: In Sect. 2, we provide a motivating example of multi-language communication service design to show our research problem, and then introduce our QoS model that can handle multiple attributes for user-centered QoS evaluation in Sect. 3. Section 4 describes the service design approach. Section 5 uses a case study to show the effectiveness of our proposed approach in language service design. Section 6 introduces some related work to elucidate the benefits of this research. Finally, the conclusion is presented in the last section.

2 Motivating Example of Language Service Design

In this chapter, we introduce a motivating example of designing a multi-language communication service, the YMC (Youth Mediated Communication)-Viet project. This project helps Vietnamese farmers obtain agricultural knowledge provided by Japanese experts [16]. Since the literacy rate of farmers in the field area is low, youths (children of the farmers) with high literacy act as mediators between the Japanese experts and the Vietnamese farmers. This project was conducted over four seasons from 2011 to 2014 in Tra On District of Vinh Long Province, Vietnam with 15–30 family of farmers in each season.

Communication between Japanese experts and Vietnamese youths in the YMC-Viet project is supported by an online multi-language tool, the YMC system [26, 27]. Vietnamese youths send field data and questions via the YMC system. The Japanese experts get the data and questions from the youths and answer the questions in Japanese via the system, which translates the replies into Vietnamese and delivers them to the youths. Therefore, the most important issue here is how to design a multi-language communication service that would make the YMC system truly effective.

To design the multi-language communication service, we used the Language Grid as the language service composition platform [11]. Due to the increase in various language resources (machine translators, dictionaries, parallel texts, etc.) for multi-language communication on the Internet, it has become possible for people to design language services to suit their own requirements [19, 25]. However, difficulties arise due to the uncertain QoS of different language services. For example, the QoS of a machine translation service is always difficult to estimate. Therefore, it is necessary to provide an approach for designing the composite services like multi-language communication services that can well handle QoS uncertainty.

In this example, the following alternative services (processes) can be used to create a multi-language communication service between Japanese and Vietnamese: (1) composite machine translation service combined with Japanese-English machine translation service and English-Vietnamese machine translation service, (2) composite Japanese-Vietnamese machine translation service combined with agriculture dictionary, (3) composite translation service combined with Japanese-Vietnamese machine translation service and Vietnamese post-editing human service, and so on. However, it is difficult to decide which composite service should be used due to the uncertain QoS of translation services as described above. Therefore, it is necessary to consider how to design an appropriate composite service to satisfy users' requirements. Moreover, it seems likely that some combination of human services and Web services will be needed, which makes the service design problem even more complicated. In this chapter, we try to address these issues by proposing a participatory service design approach based on user-centered QoS [3].

3 QoS Model for Service Design

3.1 QoS Attributes

QoS models have been widely discussed in the area of services computing. Zeng et al. proposed a multidimensional QoS model for service composition, which considers several non-functional service attributes like cost, response time, availability, reputation and so on [31]. Some previous work also considered domain-specific QoS attributes [4, 23]. To design services for the real world, it is always necessary to consider both generic non-functional QoS attributes like response time and cost, and domain-specific QoS attributes like translation quality. To achieve our goal, this work introduces an extension of previous QoS models.

We start with the definition for QoS attributes as adopted by most previous studies, i.e., the ith QoS attribute of service s is written as $q_i(s)$. We use $Q(s) = (q_1(s), q_2(s), \ldots, q_n(s))$ to represent the vector of QoS values of service s.

To design complicated services in the real world like multi-language communication services, it is not appropriate to compute the average QoS values due to uncertainty in the QoS values. Therefore, we use quality ranges to describe the QoS attributes. Mohabbati et al. proposed the approach [24] of aggregating the quality range of a composite service from those of atomic services, and this is used in our research, i.e., the quality range of the ith QoS attribute of service s is defined as $q_i^R(s) = [q_i^{LB}(s), q_i^{UB}(s)]$, where $q_i^{LB}(s)$ and $q_i^{UB}(s)$ are lower and upper bound values of the QoS attributes, respectively. There is a disadvantage in using quality ranges to handle QoS uncertainty. When the volume of QoS data becomes huge, the quality range will become large. In that case, the quality range becomes ineffective as a basis for designing a composite service. This disadvantage can be addressed by analyzing the distribution of QoS data as well as the quality range.

Since different QoS attributes have different computation methods, it is necessary to normalize different QoS attributes when using quality ranges to aggregate multiple QoS attributes. We normalize QoS attributes by using the min-max Equation (1).

$$q_i'(s) = \begin{cases} \frac{q_i(s) - min(q_i(s))}{max(q_i(s)) - min(q_i(s))} & \text{if QoS attribute is positive} \\ \frac{max(q_i(s)) - q_i(s)}{max(q_i(s)) - min(q_i(s))} & \text{if QoS attribute is negative} \end{cases} \tag{1}$$

where $max(q_i(s))$ and $min(q_i(s))$ are the pre-defined maximum value and minimum value of QoS attribute $q_i(s)$, which are different from the lower and upper bound values of the QoS attributes obtained from the historical data. Positive QoS attributes represent valuable attributes that should be increased, e.g. translation quality. Negative attributes represent a type of penalty and should be reduced, e.g. the cost of invoking a service. The quality range of the ith QoS attribute of service s after normalization can be described as $q_i'^R(s) = [q_i'^{LB}(s), q_i'^{UB}(s)]$.

The min-max normalization approach may not be applicable to some QoS attributes with high variation with a skewed distribution; take as an example a Poisson distribution, it could have a large max value but the mean may be relatively small. Some previous work focuses on how to solve the normalization methods of QoS attributes [29], but this is not the focus of this chapter.

3.2 User-Centered QoS Evaluation

We define the QoS requirements over m ($m \leq n$) attributes from a user as $C(s) = (c_1(s), c_2(s), \ldots, c_m(s))$, where $c_i(s)$ is the ith constraint of the QoS attribute from the user over service s. Moreover, we use $P(s) = (p_1(s), p_2(s), \ldots, p_m(s))$ to define whether the QoS constraint is satisfied or not. For each QoS constraint $c_i(s)$, $p_i(s)$ is calculated using Eq. (2).

$$p_i(s) = \begin{cases} 1 & \text{if } c_i(s) \text{ is satisfied} \\ 0 & \text{if } c_i(s) \text{ is not satisfied} \end{cases} \tag{2}$$

We use $w_i(s) \in [0, 1]$ to define the weight of $c_i(s)$ that is decided by user priority of QoS constraints, where $\sum_{i=1}^{m} w_i(s) = 1$.

To evaluate user satisfaction, we use two utility functions: feasibility $Utility_F(s)$, and optimity $Utility_O(s)$, which was proposed in our previous work [21].

Utility of feasibility is used to evaluate whether the service composition is feasible based on user requirements, and is calculated by Eq. (3). $Utility_F(s) = 1$ means that all the QoS constraints from the user are satisfied. Otherwise, one or more constraints are not satisfied.

$$Utility_F(s) = \sum_{i=1}^{m} p_i(s) \cdot w_i(s) \tag{3}$$

Utility of optimity is used to evaluate whether the service composition is optimal based on user requirements, and is calculated by Eq. (4). $Utility_O(s)$ is calculated when $Utility_F(s) = 1$ and it is an aggregation of the QoS values of all attributes. $w_i'(s) \in [0, 1]$ is the weight of $q_i'(s)$ that is decided by user priority of QoS attributes, where $\sum_{i=1}^{n} w_i'(s) = 1$.

$$Utility_O(s) = \sum_{i=1}^{n} q_i'(s) \cdot w_i'(s) \tag{4}$$

In previous work, Alrifai et al. proposed several types of relations between QoS attributes: independent, correlated, and anti-correlated [2], which is also useful in our research. With the independent type, the values of two QoS attributes are independent of each other, e.g. the QoS attributes of availability and translation quality in a translation service. With the correlated type, a service that is good in one attribute is also good in the other attribute. In the anti-correlated type, there is a clear trade-off between the two attributes, e.g. the QoS attributes of translation quality and cost in a translation service. In the case of anti-correlated type, *Utility of optimity* is affected by the weights yielded by the users' various requirements.

4 Service Design Process

Considering QoS uncertainty, combination of human services and Web services, and users' various requirements, it is important to design composite services by using iterative evaluation and simulation of the QoS before the service is implemented and used in the real world. First, it is natural to evaluate users' satisfaction throughout the whole design process. Moreover, when designing hybrid composite services by combining Web services and human services, it is necessary to evaluate the quality of each hybrid service by simulation. Therefore, we propose a participatory service design approach for addressing these issues. Participatory design has been proposed for community informatics [6] and multiagent systems [13], which is new in service-oriented computing especially in the context of user-centered QoS for service composition. The process of service design includes four steps.

(1) *Composite service generation*: Composite service for participatory design is generated (created or refined) based on existing QoS data by applying the following process. First, predicted utility of feasibility $Utility_F^p(s_i)$ of each candidate composite service is computed based on the upper bound values of the QoS attributes and user requirements. If the candidate is feasible ($Utility_F^p(s_i) = 1$), predicted utility of optimity $Utility_O^p(s_i)$ is then computed. The candidate service with the largest utility of optimity is then subjected to a participatory simulation.

(2) *Participatory simulation*: In the participatory simulation, human services are simulated by human participants, whose features are selected or trained to be similar to the human services in the composite service. Participatory simulations are useful for controlled experiments because they make it easy to prepare the environment for testing [13]. To conduct the participatory simulation, the service designer does not have to create a real system. Instead, all interactions between the computer (services) and human are described using simple scenario description languages like *Q* Language [10]. We extended the *Q* Language to develop a participatory simulation tool for service design, on which the simulation scenario can be conducted.

(3) *QoS evaluation*: In the participatory simulation, the log data of the participants can be used for analysis and refinement of the service model. Therefore, the values of QoS attributes can be obtained from the execution logs of the simulation, and used for interactive service model refinement. In this step, if the evaluated QoS can satisfy users' requirements, real-world experiments (field application) can be conducted. Otherwise, composite service is refined.

(4) *QoS data update*: QoS data will be updated based on results of the participatory simulation for further user-centered service modeling and refinement. Participatory design is iterated until the simulation result satisfies user requirements.

5 Case Study

5.1 Experiment, Result and Analysis

To illustrate the effectiveness of our proposed approach for service design, especially in the hybrid service composition environments involving Web services and human services, we conducted a field experiment (YMC-Viet project) to design a multi-language communication service for supporting Vietnamese farmers as was mentioned in Sect. 2.

Services for composition. In realizing the multi-language communication service for the YMC-Viet project, many composite services were available, e.g. human translation, parallel text service, machine translation service combined with dictionary, and so on. All of these composite services can be created based on atomic services. Table 1 shows a part of the Web services provided by the Language Grid and human services as used in case study.

Table 1 List of web services and human services for multi-language communication service design in the YMC-Viet experiment

Service	Service type	Description
s_1	Composite web service	Composite Japanese-Vietnamese machine translation service combined with agriculture dictionary
s_2	Composite web service	Composite Japanese-English machine translation service combined with agriculture dictionary
s_3	Composite web service	Composite English-Vietnamese machine translation service combined with agriculture dictionary
s_4	Atomic web service	Japanese-Vietnamese parallel text service for agriculture
h_1	Human service	Japanese pre-editing service
h_2	Human service	English post-editing service
h_3	Human service	Vietnamese post-editing service
h_4	Human service	Japanese-English human translation service
h_5	Human service	Japanese-Vietnamese human translation service

QoS attributes and QoS data. QoS in the language service domain consists of non-functional QoS attributes (translation cost, execution time, etc.) and functional QoS attributes (translation quality). In our experiment, cost, execution time and translation quality are used as the dominant QoS attributes. Since there is no QoS data available before the field experiments, the QoS ranges of the different composite services were created based on the simulations using the same conditions as the YMC-Viet project.

User requirements. The user requirements for QoS attributes of the multi-language communication service are $q'_{translation\_quality}(s) > 0.7$ (weight is 0.2) and $q'_{cost}(s) > 0.5$ (weight is 0.8) after normalization. Since the QoS attributes of cost and execution time are related to each other, we do not take the attribute of execution time as one of the user requirements.

We use the participatory service design approach proposed in this chapter for designing the multi-language communication service in the first two seasons' experiments. Table 2 shows the iterative participatory design result (from process P1 to P5). Since the parallel text service is used from process P2 to P5, we omit it from the Table 2. Figure 2 shows details of the quality ranges and QoS values in each process described in Table 2. Moreover, the refinement process of composite service design is also shown. Four times of refinement were observed in the experiment: from P1 to P2, from P2 to P3, from P3 to P4, and from P4 to P5. Finally, composite service P5 satisfied the user requirements and was the best of all possible composite services. Accordingly, it was used as the composite service model for implementation of the multi-language communication tool (YMC system) for the field experiment after the second season in 2012. The refinement of composite services is described in detail in [17].

Table 2 Composite service processes for participatory simulation used in the YMC-Viet experiments and QoS evaluation result of different QoS attributes

Composite service	Service workflow	QoS value		
		Cost (USD)	Time (min)	Translation quality
P1	s_1	0.10	0.01	1.36/5.00
P2	$h_1 \rightarrow s_1$	1.33	10	2.14/5.00
P3	$h_1 \rightarrow s_1 \rightarrow h_3$	3.33	20	2.67/5.00
P4	$h_4 \rightarrow s_3 \rightarrow h_3$	25.18	102	4.40/5.00
P5	$h_1 \rightarrow s_2 \rightarrow h_2 \rightarrow s_3 \rightarrow h_3$	17.51	77	4.28/5.00
P6	h_5 (for comparison)	44.00	150	5.00/5.00

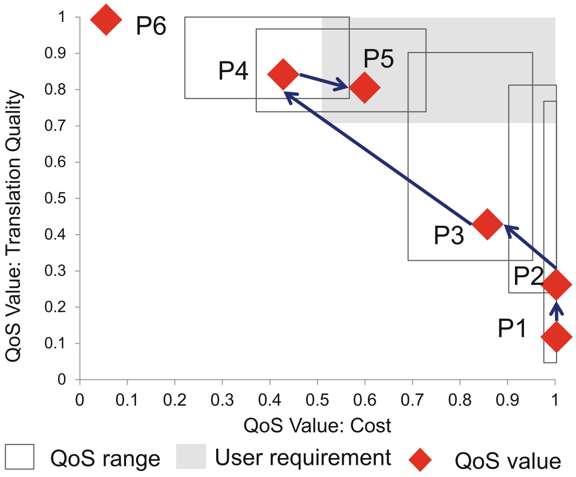

Fig. 2 Change of composite services and QoS values with participatory service design (P = Process)

5.2 *Discussion*

The final service model (Process P5) worked well in the field experiment. However, there are several issues that should be considered in the future.

First, it is effective but not efficient to use just quality range in refining service model because quality range is large when a lot of data are collected and cannot always reflect the prediction as described in Sect. 3. Note that four times of refinement were needed as all composite service candidates had the possibility (their upper bound) to satisfy the user requirements, see Fig. 2. A simple heuristic is to consider the distribution of QoS values within each range. For example, most of the QoS values of process P1 are quite low, so it seems reasonable to drop process P1 from further consideration.

Second, some participatory simulations yielded results that did not improve the user requirements. There are two possible reasons: one lies in the uncertainty of the QoS data, and the other is the human participants in the simulation did not provide an adequate view of reality. Therefore, it is also necessary to support virtual participatory simulations with real-world experiments.

Finally, the service design problem becomes more complicated as the number of anti-correlated QoS attributes increases, which makes it difficult to satisfy the user requirements. In that case, the service workflow optimization approach should be introduced, as we have shown it to be effective in designing crowdsourcing workflows for complicated tasks [8].

6 Related Work

Research into traditional QoS-aware service composition problems, always makes the assumption that composite services are given beforehand. The main research issue is then to select the most appropriate set of atomic services based on QoS optimization. Zeng et al. [31] proposed a multidimensional QoS model for Web service composition and several optimization approaches for service selection in both static and dynamic environments. Liu et al. described in detail the computation process for QoS selection, including normalization, and weighted aggregation [22]. Wang et al. described not only the processes of normalization and weighed aggregation, but also a QoS ontology description [28]. The process of filtering Web service according to user preference was described by Cao et al. [5]. Huang et al. described QoS aggregation for different workflow patterns in a composition service [9]. To deal with the dynamic service selection problem, QoS prediction has been studied including approaches that use historical data, user experience, and so on [32, 33].

The biggest difference between our research and previous research is that we deal with designing composite services in the real world rather than selecting atomic services for a given composite service. Moreover, most of the above work neglected the difficulties of handling QoS issues in real service composition environments. First, there are situations that some QoS attributes cannot be aggregated for composite services. For example, it is difficult to compute the translation quality of a composite translation service by simply aggregating its component atomic service (machine translation service, morphological analysis service, dictionary service, etc.). Second, when there are multiple QoS attributes for services, it is always difficult to maximize all the QoS attributes because some may have anti-correlated relations. Third, QoS values always vary with the context for different types of service invocation, which has been pointed out as QoS uncertainty [30]. Therefore, it is important to evaluate QoS uncertainty and consider user requirements when designing composite services, which is the focus of this chapter.

In recent years, combining human services with Web service has become a new important issue with the development of crowdsourcing and cloud computing environments. Kern et al. proposed a statistical quality control approach for human services and validated their approach using crowdsourcing experiments [14]. Efforts to deal with human services in crowdsourcing environments can be also found in some other works [15, 18]. To design multi-language communication (or translation) services, attempts have been to combine human services and Web services [18] for ensuring translation quality. However, most previous studies focused on the quality control of human services and the composition mechanisms of hybrid services. It is extremely important to test the service composition environments for human-computer interaction and study human behaviors. In this chapter we use the participatory service design approach to deal with the QoS uncertainty issues and human-computer interaction issues.

7 Conclusion

Language service design is a typical problem tackled by QoS-aware service composition since it deals with how to design composite services considering the uncertainty of services in the real world and how to meet the requirements of various users. In this chapter, we proposed a service design approach based on user-centered QoS using a case study of designing language services for a multi-language communication experiment. We first extended the QoS model proposed in previous work to address the uncertainty of language services and user requirements by defining two types of utilities to compute user-centered QoS. Then, we proposed the participatory service design process that consists of composite service generation, participatory simulation, QoS evaluation and QoS data update. Finally, we used a field study of multi-language communication service design to illustrate the effectiveness of our proposed design approach.

Acknowledgements This research was supported by a Grant-in-Aid for Scientific Research (A) (17H00759, 2017-2020) and a Grant-in-Aid for Young Scientists (A) (17H04706, 2017-2020) from Japan Society for the Promotion of Science (JSPS).

References

1. Alrifai, M., Risse, T.: Combining global optimization with local selection for efficient QoS-aware service composition. In: Proceedings of the 18th International Conference on World Wide Web, pp. 881–890. ACM (2009)
2. Alrifai, M., Skoutas, D., Risse, T.: Selecting skyline services for QoS-based web service composition. In: Proceedings of the 19th International Conference on World Wide Web, pp. 11–20. ACM (2010)
3. Bramantoro, A., Ishida, T.: User-centered QoS in combining web services for interactive domain. In: Fifth International Conference on Semantics, Knowledge and Grid, 2009. SKG 2009, pp. 41–48. IEEE (2009)

4. Canfora, G., Penta, M.D., Esposito, R., Perfetto, F., Villani, M.L.: Service composition (re)binding driven by application-specific QoS. In: Proceedings of the 4th ICSOC, pp. 141–152 (2006)
5. Cao, J., Huang, J., Wang, G., Gu, J.: QoS and preference based web service evaluation approach. In: Eighth International Conference on Grid and Cooperative Computing, (GCC 2009), pp. 420–426 (2009)
6. Carroll, J.M., Rosson, M.B.: Participatory design in community informatics. Des. Stud. **28**(3), 243–261 (2007)
7. Cavallo, B., Di Penta, M., Canfora, G.: An empirical comparison of methods to support QoS-aware service selection. In: Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems, pp. 64–70. ACM (2010)
8. Goto, S., Ishida, T., Lin, D.: Understanding crowdsourcing workflow: modeling and optimizing iterative and parallel processes. In: Fourth AAAI Conference on Human Computation and Crowdsourcing, pp. 52–58 (2016)
9. Huang, A.F.M., Lan, C.W., Yang, S.J.H.: An optimal QoS-based web service selection scheme. Inf. Sci. **179**, 3309–3322 (2009)
10. Ishida, T.: Q: A scenario description language for interactive agents. Computer **35**(11), 42–47 (2002)
11. Ishida, T. (ed.): The Language Grid: Service-oriented Collective Intelligence for Language Resource Interoperability. Springer Science & Business Media, Berlin (2011)
12. Ishida, T.: Intercultural collaboration and support systems: a brief history. International Conference on Principle and Practices in Multi-Agent Systems (PRIMA 2016). Invited paper, pp. 3–19 (2016)
13. Ishida, T., Nakajima, Y., Murakami, Y., Nakanishi, H.: Augmented experiment: participatory design with multiagent simulation. In: Proceedings of the 20th international joint conference on Artifical intelligence, pp. 1341–1346. Morgan Kaufmann Publishers Inc., Burlington (2007)
14. Kern, R., Thies, H., Satzger, G.: Statistical quality control for human-based electronic services. Serv.-Oriented Comput. 243–257 (2010)
15. Khazankin, R., Psaier, H., Schall, D., Dustdar, S.: QoS-based task scheduling in crowdsourcing environments. Serv.-Oriented Comput. 297–311 (2011)
16. Lin, D., Ishida, T.: Participatory service design based on user-centered QoS. In: Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 01, pp. 465–472. IEEE Computer Society (2013)
17. Lin, D., Ishida, T.: User-centered service design for multi-language knowledge communication. In: Serviceology for Services, pp. 309–317. Springer, Berlin (2014)
18. Lin, D., Ishida, T., Murakami, Y., Tanaka, M.: Improving service processes with the crowds. In: Service-Oriented Computing-ICSOC 2011 Industry Track, pp. 295–306. Springer, Berlin (2012)
19. Lin, D., Murakami, Y., Ishida, T., Murakami, Y., Tanaka, M.: Composing human and machine translation services: language grid for improving localization processes. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation, pp. 500–506 (2010)
20. Lin, D., Murakami, Y., Tanaka, M.: Designing dynamic control mechanisms for service invocation. J. Inform. Process. **19**, 52–61 (2011)
21. Lin, D., Shi, C., Ishida, T.: Dynamic service selection based on context-aware QoS. In: 2012 IEEE Ninth International Conference on Services Computing (SCC), pp. 641–648. IEEE (2012)
22. Liu, Y., Ngu, A.H., Zeng, L.Z.: QoS computation and policing in dynamic web service selection. In: Proceedings of the 13th international WWW Alt, pp. 66–73. ACM, New York (2004)
23. Ma, Q., Wang, H., Li, Y., Xie, G., Liu, F.: A semantic QoS-aware discovery framework for web services. In: IEEE International Conference on Web Services (ICWS 2008), pp. 129 –136 (2008)
24. Mohabbati, B., Gašević, D., Hatala, M., Asadi, M., Bagheri, E., Bošković, M.: A quality aggregation model for service-oriented software product lines based on variability and composition patterns. Serv.-Oriented Comput. pp. 436–451 (2011)

25. Murakami, Y., Lin, D., Ishida, T.: Service-oriented architecture for interoperability of multilanguage services. In: Towards the Multilingual Semantic Web, pp. 313–328. Springer, Berlin (2014)
26. Takasaki, T., Mori, Y., Ishida, T., Otani, M.: Youth mediated communication: knowledge transfer as intercultural communication. Serv. Comput. Lang. Res. (2017)
27. Takasaki, T., Murakami, Y., Mori, Y., Ishida, T.: Intercultural communication environment for youth and experts in agriculture support. In: 2015 International Conference on Culture and Computing (Culture Computing), pp. 131–136. IEEE (2015)
28. Wang, X., Vitvar, T., Kerrigan, M., Toma, I.: A QoS-aware selection model for semantic web services. In: Proceedings of the 4th ICSOC, pp. 390–401. Springer, Berlin (2006)
29. Yau, S.S., Yin, Y.: QoS-based service ranking and selection for service-based systems. In: Proceedings of the 2011 IEEE International Conference on Services Computing (SCC 2011), pp. 56–63. Washington, D.C. (2011)
30. Yu, Q., Bouguettaya, A.: Computing service skyline from uncertain qows. IEEE Trans. Serv. Comput. **3**(1), 16–29 (2010)
31. Zeng, L., Benatallah, B., Ngu, A.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for web services composition. IEEE Trans. Softw. Eng. **30**(5), 311–327 (2004)
32. Zeng, L., Lingenfelder, C., Lei, H., Chang, H.: Event-driven quality of service prediction. Serv.-Oriented Comput.-ICSOC **2008**, 147–161 (2008)
33. Zheng, Z., Ma, H., Lyu, M.R., King, I.: Qos-aware web service recommendation by collaborative filtering. IEEE Trans. Serv. Comput. **99**, 140–152 (2011)