Apostolos Papageorgiou, André Miede, Dieter Schuller, Stefan Schulte, Ralf Steinmetz: *Always Best Served: On the behaviour of QoS- and QoE-based Algorithms for Web Service Adaptation*. In: Proceedings of the eighth IEEE International Workshop on Managing Ubiquitous Communications and Services (PerCom Workshops 2011 - MUCS), p. 71-76, March 2011, ISBN 978-1-61284-936-2.

Always Best Served: On the behaviour of QoS- and QoE-based Algorithms for Web Service Adaptation

Apostolos Papageorgiou, André Miede, Dieter Schuller, Stefan Schulte, Ralf Steinmetz

Technische Universität Darmstadt
Multimedia Communications Lab - KOM
Darmstadt, Germany
{*apostolos.papageorgiou, andre.miede, dieter.schuller, stefan.schulte, ralf.steinmetz*}*@kom.tu-darmstadt.de*

***Abstract*—One of the primary issues in pervasive computing is the adaptation of the communication to the needs of limited devices. Web services are one of the technologies that present both big challenges and big potentials when it comes to their adapted usage in pervasive systems, because they carry communication overhead in order to support interoperability and platform-independence through self-description. In this paper it is explained how Web service communication can be adapted for limited devices and why it is important to choose intelligently among a variety of adaptation mechanisms, based on the system context. Further, two algorithms (one Quality of Service-based and one Quality of Experience-based) for the decision support of the mentioned problem are provided in order to measure and discuss the impact that the peculiarities of the problem have on their behaviour.**

***Keywords*-Web Services; Pervasive Computing; QoS; QoE;**

I. Introduction

Since the birth of pervasive computing, the adaptation of the communication in order to enhance the Quality of Service (QoS) of applications has been one of the biggest concerns in the field [7]. Such adaptations can be performed on different layers, e.g., on the communication channel layer, such as in the much investigated "Always Best Connected" (ABC) issue, or on the application layer, as it is done during Web content adaptation. Another possibility appears on the service layer, where the protocol (or the access method) that is used to communicate with particular services is adjusted to the system context. Limited devices have already appeared as participants of Service-oriented Architectures (SOA), mostly as simple Web service consumers.

Constraints such as limited bandwidth, CPU, memory, or energy resources, in combination with the communication overhead introduced by Web service standards (WSDL, SOAP), can lead to unacceptable QoS. Despite technological progress, the gap will continue to exist. The latest analyses of future wireless technologies strengthen this argument. In the book of Sesia et al. [10] about LTE (Long Term Evolution of 3G mobile networks), five categories of user equipment are defined. According to it, devices of higher categories will be able to use connection rates up to six times greater than those of lower categories. Of course, the wired connections of the future will be even faster than that. Furthermore, devices less capable than smartphones, such as sensor nodes, will be able to consume Web services. So, the big differences in device capabilities and connection qualities will maintain the need for adaptation, as the size of the data that is processed and wirelessly transmitted is growing parallel to all other technological developments [2].

For this reason, adaptation mechanisms for Web services have appeared. An adaptation mechanism means the re-offering of a Web service with a different protocol or access method, e.g., Compressed SOAP, RMI, SOAP-over-UDP, REST. In our recent survey [8], most important mechanisms were analyzed. In the evolving Internet of Services (IoS), the most obvious solution for adapting Web services without access to the provider system is through the generation of proxies in mediation layers [1]. Approaches for dynamic service re-deployment (as in [6]) could be considered on the provider side, but are not applicable without access to the provider system. A research question that arises is how the mediation layer decides which adaptation mechanism(s) to use for each service. Two main general approaches exist with respect to the triggering of the adaptation:

- *QoS-based adaptation*: the adaptation actions are initiated based on objective values of technical parameters.
- *QoE-based (Quality of Experience) adaptation*: the adaptation actions are initiated based on past user decisions or user feedback.

The purpose of the work at hand is to investigate the application and the limits of QoS and QoE in this Web service adaptation scenario. The results should indicate which of the two should be preferred, depending on the information that is available to the mediation layer. Although this evaluation is our main contribution, the description of the "Always Best Served" problem and of the two algorithms (one QoS-based and one QoE-based) for its solution consist further contributions. Thus, we explore related work in Section 2 and describe our exact scenario in Section 3. The two algorithms that we have developed are described in Section 4, while they are evaluated and discussed in Section 5.

II. Related Work

Many researchers in the field of pervasive computing are familiar with the ABC issue [5], [12]. The goal of ABC is to let the device switch among many possible access networks (e.g., WLAN, GSM, UMTS, Bluetooth, and other variants from the 802.11 and 802.15 "families", while new technologies such as LTE are also coming into play). The corresponding selection problem is often modelled and handled as a knapsack problem (NP-hard) [4], while QoE-based approaches have also appeared [3]. However, an issue similar to ABC appears if we move up in the OSI model, from the network to the transport and session layers. There, adapted Web service access methods have to be examined and selected, as described in the introduction. In accordance with ABC, we introduce the term "Always Best Served" (ABS) and describe it in Section 3. Despite similarities to ABC, ABS appears on a different level (needs partly different context), is less deterministic (conditions that match each of the alternatives have not been researched in such detail), and the technologies that make the issue arise had been immature till now.

A relevant and detailed analysis of QoS- and QoE-based optimizations for mobile technologies can be found in [11], where general theoretical issues, as well as specialized algorithms for particular problems are examined. Still, [11] examines the two approaches separately, does not provide comparative analyses, and does not focus on ABS-specific attributes, such as the characteristics of Web services. Furthermore, decision-making algorithms depend on the scenario. [11] presents solutions for problems such as routing or mobile network (re-)configuration, but the ABS issue has differences, e.g., concerning the context structure and the types of knowledge incompleteness that may appear. For example, ABC-algorithms (but also routing algorithms) take decisions for a given user/session, thus they are less concerned with missing information about the networks and devices of other users.

III. The "Always Best Served" Scenario

The new scenario (ABS) is described in the following. First the background and then a more detailed technical description and a visualization are provided.

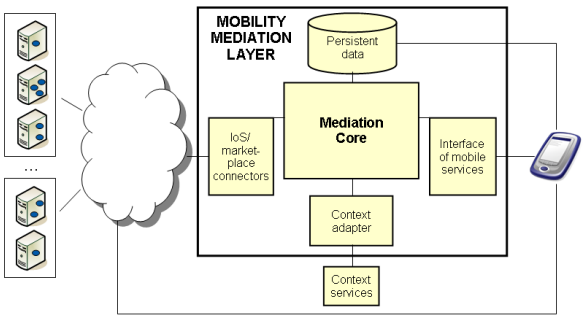
A. Definitions, Assumptions, Research Question

A big set of Web services offered by different providers through service marketplaces and a big set of client devices with different features co-exist in the IoS. A *mediation layer* is an extension of the service marketplace that can apply different adaptation mechanisms in order to offer the existing services through new interfaces, with different protocols or access methods. An *adaptation mechanism* can be abstractly defined as the generation of a proxy that runs on the mediation layer. The *protocols or access methods* refer to the different ways with which the provider and the client device can communicate. Some examples have been mentioned in the introduction. The enforcement of the adaptation mechanisms (i.e., the generation of proxies) results in the existence of different *alternatives* for the consumption of the same Web service, among which the clients can choose. Two questions arise: *a)* How does the mediation layer decide which proxies to generate? *b)* How do the devices decide which proxy to use (if any)? We examine the issue from the *mediation layer point-of-view*, so we focus only on question *a*. Furthermore, decisions of the devices (question *b*) are slightly simpler and need therefore less decision support, because the decision depends mostly on the current situation and does not necessarily need to consider information from other calls or other users.

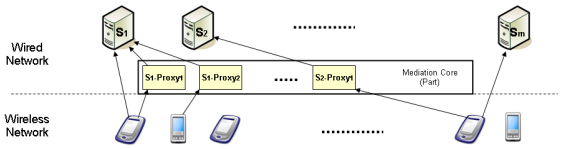
A *basic assumption* for this decision support is that the mediation layer has limited capacity and/or is concerned with security. Otherwise it could generate all possible proxies. This assumption can be understood on the basis of example adaptation mechanisms: In order to re-offer a particular service with, for example, RMI or SOAP-over-UDP, the mediation layer has to deploy new modules in the RMI registry or in its Web container, respectively. Every new proxy needs some resources and inevitably means the opening of new interfaces/ports on the mediation layer. So, it is obvious that not every possible proxy can be generated for the infinite number of services that are to be found in the IoS. Ideally, the decision support appears in the form of a *scoring algorithm* that determines how adequate an adaptation mechanism (i.e., its corresponding proxy) would be for each known service under the current conditions. Because the quality of such an algorithm cannot be objectively evaluated, we formulate a more specific research question, which will be answered in our evaluation: "How do two important scoring algorithms, namely one QoS-based and one QoE-based, behave in the context of ABS?".

B. Scenario and Architecture

In the context of a project for the IoS, we are developing the Mobility Mediation Layer (MML). As abstractly shown in Fig.1a, the goal of the MML is to provide limited or mobile clients with an interface to services which are hosted on various external providers and are made available through global service marketplaces, while mediating the service consumption by performing various adaptation mechanisms. Although the MML has further capabilities (e.g., automatic context enrichment), we focus here on the overhead reduction through the generation of proxies, as previously described. An example where a service (S_1) can be consumed either directly or through one of two different proxies is shown in Fig.1b. To implement this, the MML uses our "Web Service Proxy Generator", a software which performs automatic code enrichments and deployment actions upon the code generated by the execution of "wsimport" for the target service.



(a) MML Overview.



(b) Web service proxies in the MML core.

Component	Device Network Connection					Device	Service			Application	
Aspect / Approach	Bandwidth	Latency	Packet loss	Stability	Disconnected periods	CPU Power	Data size / SOAP size	SOAP message size	Processing time	Service call frequency	Service call dependence
SOAP-over-UDP		≥m	≤s					≤m	≤s		≤s
Compression	≤s					≥m		≥m			
…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…

(c) Characteristics of the possible adaptation mechanisms. The complete table can be found in [8]

Figure 1. Important aspects of the "Always Best Served" scenario.

The question may arise: "Why not generate the same type of proxy for all services?". The answer is that there is no "best adaptation mechanism", which would fit all cases. After a detailed study, we found out that different access methods should be preferred under different conditions. These conditions concern the network, the client device, the service itself, and more. We analyzed the possible access methods according to the conditions under which they are expected to achieve their maximum benefit and provided the results in [8]. Fig.1c shows an excerpt of this result. For example, as seen in the table at the bottom, a compression proxy is adequate when the bandwidth is *small*(s), the CPU power of the device is *medium*(m) or *high*(h), and the message sizes of the Web service are also *medium* or *high*. The lines of this table already look like rules for the generation of proxies, but they are far from being deterministic deterministic for the decision process, as other factors may come into play, such as weighting, different goals or utility functions, user feedback etc. Thus, different algorithms can be built on that foundation.

IV. Decision Support Algorithms for the ABS

A view of the monitored system context is shown in Table I. Every row is the record (or *log*) of one particular service call. This corresponds with the realistic monitoring

Table I
Monitored system context from past Web service calls

	Connection				Dev.	App.		Service			
Record	Bandwidth (q_1)	Latency (q_2)	Packet Loss (q_3)	Stability (q_4)	CPU Power (q_5)	Call frequency (q_6)	Call dependence (q_7)	Message size (q_8)	Overhead ratio (q_9)	Processing time (q_{10})	Chosen proxy (q_{11})
r_1	u	m	s	u	u	u	s	s	s	m	p_4
r_2	u	u	h	u	h	u	m	s	m	h	p_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

capabilities of modern service-oriented systems. The representation of the monitored data is not critical (u stands here for *unknown*, while s, m, and h have been already explained). The last column (q_{11}) indicates which access method (proxy) has been chosen by the user for this call.

So, as their definitions imply (see Introduction), a QoS-based approach would rate each proxy by comparing its "ideal technical conditions" (Fig.1c) with the monitored service usage pattern, while a QoE-based approach would do the work by analyzing the relationships between the value of q_{11} and the values of the other attributes. The latter analysis would indicate "how users decide". The details of the algorithms may affect the results, but the main differences in the evaluation results will be a consequence of the conceptual difference between QoS and QoE.

A. The QoS-based Algorithm

As it is usually the case in QoS-optimization theory, the proposed QoS-based algorithm uses a utility function for the scoring of the different options. Our utility function is based on vector distances. Every service call record (row of Table I) is represented as a vector. The "distance" between this vector and each "proxy reference vector" (row of Fig.1c) is measured. The latter represents the optimal service calls for the proxy and comes from the analysis of [8], which is summarized in a table, as shown in Fig.1c. This decision was driven by the fact that both service call records and proxy records are already in a vector-like form with ordered symbols (s, m, h) as values. This makes their distance-based comparison simple and meaningful.

Thus, a utility function $n_{p_i}(r_j)$ ($n : P \times R \to \mathbb{R}$, where P is the set of proxies and R is the set or records) attaches a score to every proxy p_i for the record r_i. The function performs a comparison for every element of the vector. For example, an h value of an attribute where the condition is $\geq m$ gives a positive difference of +1, while an s value gives a negative difference of -1. These differences might also be "weighted". As this is done for each service call record of the examined service, the results are aggregated to a total score of each proxy for the service.

The aggregation is performed as follows: Let $R_{s_k} \subseteq R$ be the set of service call records of service s_k and N_{s_k,p_i} be the *set of the results* of the application of the utility function $n_{p_i}(r)$ with $r \in R_{s_k}$. Let

$$N^+_{s_k,p_i} := \{x | x \in N_{s_k,p_i} \wedge x > 0\} \text{ and}$$
$$N^-_{s_k,p_i} := \{x | x \in N_{s_k,p_i} \wedge x \leq 0\}$$

then $\rho^+ := \frac{|N^+_{s_k,p_i}|}{|N_{s_k,p_i}|}$ is the quota of calls that would result in a positive score for this proxy.

Let $\gamma := [0..1]$ be the minimum acceptable threshold for the positive score quota. On the basis of γ, weights $w^+(\gamma, \rho^+)$ and $w^-(\gamma, \rho^+)$ are calculated for the aggregation (formulas are omitted), so that the final scoring function is

$$f(p_i, s_k) := w^+(\gamma, \rho^+) \sum N^+_{s_k,p_i} + w^-(\gamma, \rho^+) \sum N^-_{s_k,p_i}$$

, whereby f ends up giving scores in the range {-2,3}. Although the generation of the proxy p_i that maximizes f for a given s_k might be reasonable, the algorithm can be used for decision support, so that the final decision may be taken by an administrator and does not have to be automatically enforced. The same holds for the QoE-based algorithm.

B. The QoE-based Algorithm

Because users have their own subjective selection criteria, a different indicator of the "proxies' suitability" is needed. In QoE-based approaches, this indicator is based on user feedback, which can be explicit or implicit. Our algorithm uses past user decisions (i.e., user choices of proxies) as implicit feedback and calculates the mentioned "proxy suitability indicator" as its probability to be selected by the users in the future, if all proxies for this service existed. As the algorithm that calculates this probability should use part of the data in order to "learn" past user behaviour and part of it in order to set evidence about the service that is examined each time, machine learning was an obvious choice. In particular, we developed an algorithm based on a Bayesian Network (BN), because BNs match our problem for two main reasons: First, they do not only classify cases (as simple decision trees do, for example), but they compute exact probabilities, as needed for a detailed proxy scoring. Second, they are an appropriate approach for setting evidences, which we have to do for every examined service.

The idea is to let the algorithm learn about a given service by examining the past user behaviour upon "similar" services which had been offered with all proxies. Two services s_1 and s_2 are similar if $q_i(s_1) = q_i(s_2), \forall i \in \{8, 9, 10\}$ (see Table I). We include in the BN the variables that are likely to determine the user selection, i.e., the variables that are most probably known to the user, e.g., $\{q_1, q_5, q_6, q_7, q_{11}\}$. Summarizing in simple steps, the following is done in order to score each proxy for a given service:

- STEP 1: A logical BN structure is manually built, showing which attributes affect the user decision.
- STEP 2: The records of similar services are analyzed to find out how users decide (generation of the Conditional Probability Tables of the BN).
- STEP 3: The records of the examined service are analyzed to find out how the service is likely to be used in the future (Evidence in the BN).
- STEP 4: Once the above has been calculated, the BN is used to infere the probability of each proxy to be used for the examined service during the next calls. This probability is the score of the proxy.

V. Experiments and Discussion

The two described algorithms have been used for initial experiments towards investigating the features and the limits of the two approaches. A direct, one-to-one comparison of the results of the two approaches would be both impossible and meaningless, because different outputs should be defined as optimal in the two cases. Instead, our idea for a meaningful evaluation was to use as reference-outputs the results of each of the algorithms in the case of complete knowledge/input. Then, each algorithm is examined concerning its behaviour (in terms of deviation to the reference-output) in different cases of *incomplete knowledge*. This idea mirrors the needs of most solutions in the field, as monitoring systems would rarely be able to provide the adaptation layer with all the useful information that we took into consideration in our analysis. Thus, the purpose of the experiments is neither to prove that the presented algorithms are optimal, nor to determine which is better, but rather to investigate their behavior. The results will then give hints about how the suitability of each approach in an own system/domain should be judged.

A. Evaluation Set

This subsection describes how the evaluation set has been initially built (before inserting any unknown values).

Sizes: We used a set of 4 possible proxies (from those analyzed in [8]) and a set of 12 services with different attributes (q_8-q_{10}). For half of the services, proxies are already provided, while the rest were the targets of adaptation. Different sizes were tested for the set of service call records.

Content and Correlations: The values of the parameters q_1-q_{11} being random, apart from the following correlations: First, q_1-q_4 get worse values with a higher probability if q_5 is *small* or *medium*, mirroring the fact that more powerful devices often use better network connections, e.g., because of WLAN-capabilities. Second, records of a particular service have a higher probability to get values that are similar to the values of the past calls of the same service. This reflects the fact that a service has a specific target group, i.e., a set of clients or client apps that use it.

User decisions: As mentioned, some of the services were offered without available proxies. For these, of course, no proxy was chosen by the user. For the rest, the chosen

Figure 2. Experimental results: Deviation of the scoring outputs for two different data set sizes and three different cases of incomplete knowledge

proxy was partly random, with the constraint that the users' choice had a probability of 80% to satisfy simple QoS-criteria, i.e., the users met a "pretty good" choice with a probability of 80%. This was a reasonable choice because of two reasons: First, it is indicated as an expected value by studies related to *users' decision accuracy* in similar fields, such as [9]. It should also be considered that the choices in our scenario are rather made by developers, or, at least, technology-aware users, depending on how and when the access method is chosen at the mobile side. Second, QoE should anyway be immediately rejected at the first place if users' choices are random or completely QoS-unaware. Thus, the QoE approach does not make much sense for much smaller levels of "QoS-awareness" of the users. However, this number may affect some results but is not critical for our further discussion, either.

The same evaluation set has been used for both algorithms. The obtained results (proxy scores) will be the baseline. These results are used as reference in order to examine the behaviour of the algorithms when inserting unknown values. For the scenarios with incomplete data, information has been only removed (and never added or changed).

B. Metrics and Setup

Both algorithms give a scoring vector for each service. This vector contains the score of each proxy for this service under the given conditions. For example, the QoS-based algorithm returned for the case of 5000 service call records with complete knowledge the vector {-0.76;-0.36;-0.57;0.45} for service s_3, thus pointing at p_4 as the most suitable proxy. After inserting u-values for the parameters q_1-q_4 in our data set (reflecting a "weak" monitoring of network connections), the algorithm returned {-0.85;-0.60;-0.63;0.29}. The Euclidean distance of these 2 vectors is used to express the deviation caused by the incompleteness of knowledge. As known, if x_i and y_i are the elements of two n-sized vectors, this distance is $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$, which gives 0.31 for the given example. Other metrics should be considered in the future.

In the selected results, we use two different sizes of the service call record set (1000 and 5000), which, combined with our two algorithms, give four different cases, corresponding with the four plots of Fig.2. For each of them, there are three different scenarios of incomplete knowledge, reflecting different cases of real monitoring systems:

- *Network-incompleteness*: ca. 25% of the knowledge about the network (q_1-q_4) is randomly set as unknown.
- *Client-incompleteness*: ca. 25% of the knowledge about the client devices and applications (q_5-q_7) is randomly set as unknown.
- *Feedback-incompleteness*: ca. 25% of the knowledge about user decisions (q_{11}) is randomly set as unknown.

The percentage of missing knowledge (25%) was chosen so that a non-trivial deviation is caused. Variations of this factor are also a subject of future work.

Thus, there are four outputs in each case (the complete plus the three incomplete scenarios). The deviation of the complete case from itself is, of course, not depicted, as it is naturally zero. This is the baseline, and it is the distance of the other scenarios from this baseline that are of interest. Further, the Feedback-incompleteness scenario is not depicted for the QoS-algorithm, because its deviation is

also zero (this algorithm does not care about user decisions). The deviations for the rest of the scenarios are shown in Fig.2 and discussed in the following. The six examined services appear on the x-axis, one next to the other.

C. Lessons Learned

As one could expect because of the definitions of the algorithms, the QoS-based approach presents bigger deviations in the case of the Network-incompleteness, while for the QoE-based algorithm the harm is bigger in the case of client-incompleteness. This is related to the nature of the problem because the proxies usually have more constraints for the network-related characteristics than for the other parameters. Contrary, clients are affected more strongly by device- or application-related parameters. The Feedback-incompleteness is a special case that affects only the QoE-based algorithm. The deviation is in this case much bigger than in any other case, although the missing information is quantitatively much less (only some missing values of q_{11}).

Two further interesting observations are related to the "deviation per service" and to the effect of the data set size. As can be seen, The deviation for the QoE-based algorithm may present slightly bigger differences between different services even inside the results of the same experiment (see, for example, s_5 and s_6 for QoE with 1000 records in Fig.2). This means that the deviation which should be expected for a service in a given scenario of incomplete knowledge may be slightly more difficult to predict in the case of QoE. Regarding the size of the data set, it is obvious that the deviations of the QoE-based algorithm get significantly smaller for bigger data sets. The QoS-based approach, on the other hand, is less dependent on the data set size.

The results would be affected by changes in the formulation of the problem or by the evaluation settings. However, the discussed insights would remain valid and they can serve as a roadmap, showing which aspects should be taken into account when deciding which approach to follow for a particular scenario of mobile Web service adaptation.

VI. Summary and Outlook

Decision support is an important part of every adaptation process. In this paper, it was examined for the case of mobile Web service adaptation for limited devices. The new problem has been introduced as "Always Best Served" (ABS). QoS and QoE were described as the two main research trends for approaching such decision support issues. In accordance to them, two algorithms for the support of ABS were described, evaluated, and discussed.

Parallel to this work, we are conducting research for the enablement of new adaptation mechanisms inside the MML. Thus, our future work is directed towards two main goals: The first goal is the development of new ways to adapt the service consumption, and the implementation of the corresponding proxy generators. The second goal is the optimization of the decision support algorithms presented in this paper, based on further problem-specific observations.

VII. Acknowledgements

This work was funded in part by means of the German Federal Ministry of Economy and Technology under the promotional reference "01MQ09016" ("Green Mobility" Project). The authors take the responsibility for the contents. Many thanks to Liang Han and Jeremias Blendin.

References

[1] M. Adacal and A. Bener. Mobile Web Services: A New Agent-based Framework. *IEEE Internet Computing*, 10(3):58–65, 2006.

[2] C. Canali, M. Colajanni, and R. Lancellotti. Performance Evolution of Mobile Web-based Services. *IEEE Internet Computing*, 13(2):60–68, 2009.

[3] K. Demestichas, A. Koutsorodi, E. Adamopoulou, and M. Theologou. Modelling User preferences and Configuring Services in B3G Devices. *Wireless Networks*, 14(5):699–713, 2008.

[4] V. Gazis, N. Alonistioti, and L. Merakos. Toward a Generic Always best Connected Capability in Integrated WLAN/UMTS Cellular Mobile Networks. *IEEE Wireless Communications*, 12(3):20–29, 2005.

[5] E. Gustafsson and A. Jonnson. Always Best Connected. *IEEE Wireless Communications*, 10(1):49–55, 2003.

[6] M. Hillenbrand, P. Müller, and K. Mihajloski. A Software Deployment Service for Autonomous Computing Environments. In *Proceedings of the International Conference on Intelligent Agents, Web Technology and Internet Commerce (IAWTIC 2004)*, 2004.

[7] K. Nahrstedt, D. Xu, D. Wichadakul, and B. Li. QoS-aware Middleware for Ubiquitous and Heterogeneous Environments. *IEEE Communications Magazine*, 39(11):140–148, 2001.

[8] A. Papageorgiou, J. Blendin, A. Miede, J. Eckert, and R. Steinmetz. Study and Comparison of Adaptation Mechanisms for Performance Enhancements of Mobile Web Service Consumption. In *IEEE World Congress on Services (SERVICES '10)*, pages 667–670. IEEE, 2010.

[9] P. Pu, P. Viappiani, and B. Faltings. Increasing User Decision Accuracy Using Suggestions. In *The 24th ACM Conference on Human Factors in Computing Systems (CHI '06)*, pages 121–130. ACM, 2006.

[10] S. Sesia, I. Toufik, and M. Baker. *LTE, The UMTS Long Term Evolution: From Theory to Practice*. John Wiley Publications, UK, 2009.

[11] D. Soldani, M. Li, and R. Cuny. *QoS and QoE Management in UMTS Cellular Systems*. John Wiley Publications, UK, 2006.

[12] C. Yiping and Y. Yuhang. A New 4G Architecture Providing Multimode Terminals Always Best Connected Services. *IEEE Wireless Communications*, 14(2):36–41, 2007.