

On personalized cloud service provisioning for mobile users using adaptive and context-aware service composition

Elarbi Badidi¹ · Yacine Atif²  ·
Quan Z. Sheng³ · Muthucumaru Maheswaran⁴

Received: 12 October 2017 / Accepted: 28 May 2018 / Published online: 13 June 2018
© The Author(s) 2018

Abstract Cloud service providers typically compose their services from a number of elementary services, which are developed in-house or built by third-party providers. Personalization of composite services in mobile environments is an interesting and challenging issue to address, given the opportunity to factor-in diverse user preferences and the plethora of mobile devices at use in multiple contexts. This work proposes a framework to address personalization in mobile cloud-service provisioning. Service personalization and adaptation may be considered at different levels, including the user profile, the mobile device in use, the context of the user and the composition specification. The user's mobile device and external services are typical sources of context information, used in our proposed algorithm to elicit context-aware services. The selection process is guided by quality-of-context criteria that combine cloud-service provider requirements and user preferences. Hence, the paper proposes an integrated framework for enhancing personalized mobile cloud-services, based on a composition approach that adapts context information using a common model of service metadata specification.

Keywords Personalization · Adaptive services · Mobile cloud computing · Composed services · Quality of context

Mathematics Subject Classification 68T35 · 68M14

✉ Yacine Atif
Yacine.Atif@his.se

¹ United Arab Emirates University, Al-Ain 15551, United Arab Emirates

² University of Skövde, 54128 Skövde, Sweden

³ Macquarie University, Sydney, NSW 2109, Australia

⁴ McGill University, Montreal H3A 0E9, Canada

1 Introduction

The past few years have witnessed a growing trend in using mobile services for practical purposes in various contexts. People of all ages use their mobile devices to access various services such as shopping on-the-go, gaming and making travel arrangements, as well as retrieving educational resources, reading newspapers, making payments, monitoring health conditions, invoking corporate or partner services, and many more. As users increasingly rely on their mobile devices for carrying out these activities, new challenges and opportunities for mobile service providers emerged as they strive to improve and enrich further the proliferation of mobile experiences for consumers.

Smartphones have become primary personal information-processing tools. Mobile users aspire further at executing a wider range of applications on smartphones. Modern smartphones' built-in sensors, such as GPS, accelerometer, proximity detector, ambient light sensor, and gyroscope are paving the way to advanced mobile applications. To facilitate these developments, mobile devices need further resources, in terms of storage and computing power, as well as enhanced software-provisioning services.

Mobile cloud computing (MCC) [1, 2] is poised to overcome substantially these limitations. MCC provides these resources from a remote computing platform such as Amazon AWS, Microsoft Azure, and Google AppEngine. As mobile devices could not process large-size applications locally, an alternative solution is to employ reusable services remotely from a cloud-hosting platform. A cloud service is typically the result of composing basic (in-house and third party) services on the fly to satisfy a user request. A personalized enhancement of this process includes an adaptation of the service presentation to the user interface and an adjustment of the service to the user context.

Personalization in cloud computing and related service provisioning processes are the emerging trends. The goal is to enable a dynamic customization of content and services that match mobile user preferences and current context. For example, a location-based service may recommend a specific store in the downtown of a city or in the duty free zone of an airport based on mobile users' location, preferences, and previous purchases. Several definitions of personalization are available in the literature. Nevertheless, we adopt Riecken's [3] and Hagen's [4] definitions respectively:

Personalization is about building customer loyalty by building a meaningful one-to-one relationship; by understanding the needs of each individual and helping satisfy a goal that efficiently and knowledgeably addresses each individual's need in a given context [3].

Personalization is the ability to provide content and services tailored to individuals based on knowledge about their preferences and behavior [4].

Personalization aims to improve mobile user experiences by adapting applications and services to meet individual preferences and contexts, which involves:

- Collecting information about the user, her/his preferences, current activities, and context information such as location and surrounding objects;
- Anticipating user requirements by making recommendations for additional services; and,

- Building trust and loyalty with mobile users to develop service satisfaction [5, 6].

Current personalization and recommendation techniques in cloud environments are based on *rule-based systems*, *content-based* and *collaborative filtering* [7–10]. However, solutions to mobile cloud service personalization expect services to adapt to the user's functional needs, the device in-use, and the user's context. Personalization in the context of MCC environments needs to answer the following questions: (a) How to adjust provisioned services to meet mobile user preferences and context? and (b) How to compose services to meet mobile users' personalized needs?

In this work, we address service personalization in the context of a cloud service composition involving in-house and third-party services, while considering user preferences and context. In doing so, we propose a framework for adaptive personalization of mobile cloud service provisioning, that makes the following original contributions:

- A definition of the various components of user-device side and cloud-service provider side, involved in the personalization process of mobile cloud service provisioning.
- A model for composing cloud services out of basic ones considering service selection, context matching, and content adaptation according to user profile, mobile device in use, and user context.
- An algorithm for the selection of context services that refresh context information about the user and her/his environment following the cloud service provider's quality of context (QoC) requirements. A similar algorithm can be used for matching services used in the composition by employing quality of service (QoS) attributes instead of QoC ones.
- An adaptation approach for service composition that takes into account the heterogeneity of third-party provided service profiles, and the heterogeneity of context information representation by third-party context services.

The novelty of this work lies in the fact that our proposed framework deals with the personalization of an adaptive dynamic composition of atomic services (in-house services and third-party services). The composition relies on the selection of appropriate basic services based on their QoS offerings. Also, the adaptation of the composed service to the current context of the user or her environment relies on the selection of suitable context services based on their QoC offerings. Furthermore, both the service composition and the adaptation process use a common description model for service offerings and context offerings respectively. These common description models allow to deal with the heterogeneity in the description of third-party services, and the heterogeneity in the description of context providers' offerings.

We have reported a preliminary version of this work in a conference on cloud computing [11]. In this paper, we enhance the conference version by emphasizing the context dimension, which was overlooked in the conference paper. A sub-section describing the concepts of context services and quality of context (QoC) extends further the background section. We also extended the related work section by providing a critical literature review of personalization in cloud-service provisioning. We updated and completely redesigned the proposed framework to take into consideration the acquisition of context from provisioned context services. We extended the adaptation engine to wrap various context representations from third-party context

service providers within a common semantic overlay. Finally, we added an algorithm for context service selection that is driven by QoC requirements of MCC providers.

The rest of this paper is organized as follows. The next section presents some background information and elaborates further on service personalization, context services' platforms, and the concept of quality of context (QoC). Section 3 describes some related work, highlights the similarities and limitations of relevant works on cloud-based service personalization, and discusses the motivation of this work. Section 4 presents an overview of the proposed framework. Section 5 describes the adaptation structures for personalized service delivery. Section 6 describes our proposed algorithm for the selection of context services based on QoC requirements of MCC service provider. Section 7 presents the process of integrating and adapting third party service offerings based on mobile user profiles and device profiles as well as context adaptation. Section 8 reveals a scenario used to evaluate our framework where the results of selecting context providers based on QoC requirements of the MCC service provider are analyzed and discussed. Finally, Sect. 9 concludes the paper with a summary of the work presented in this paper, and highlights some future work.

2 Background

2.1 Service personalization

Personalization of services requires collecting information while interacting with the user. The service provider uses this information to deliver appropriate content and tailored services that meet the user's needs. The broad business goals of personalization are [12]:

- To serve the user (customer) in an efficient way by anticipating her/his needs.
- To interact efficiently with the user and satisfy mutual goals of both parties (customer and service provider).
- To build a relationship with the customer to encourage subsequent purchases.

User satisfaction is the ultimate goal of service personalization. Meeting user's needs leads to trust-building between users and service-providers. This relationship will result in reusing delivered services and recommending them to user's social circles and acquaintances. Personalization of services relies on collected information about the user. The user might specify her/his needs explicitly during the registration to the service. Besides, the history of the user's previous sessions, actual interactions with services, and current context information represent an important source of information from which the service provider can infer a user profile and anticipate his/her needs. Using predefined rules and techniques such as collaborative filtering, service providers could determine some suitable contents to recommend to specific users.

Personalization is not a new concept in the service computing landscape, but it is certainly going to be a crucial aspect for the success of MCC services. Nevertheless, service providers have to understand that motivating consumers to embrace a service is not as significant as delivering a personalized experience that meets their implicit and explicit needs. Today's consumers require services that can provide them with tailored and relevant information. They are also increasingly expecting services that

can adapt to their current context such as location and current activities, while satisfying their functional and nonfunctional requirements. Therefore, mobile service providers need to provide the degree of personalization of their services to retain and enlarge their customers' base. To achieve this degree of personalization, MCC providers need to accumulate sufficient information about users' preferences and their context, to customize and recommend pertinent services.

Mobile users are generally less willing to enter lots of data. Therefore, the more information that is implied from the mobile user's profile, context, and data retrieved from the mobile device, the more likely that user is willing to engage further in utilizing MCC services. However, this may raise privacy concerns, unless users are opted to share information with MCC provider under a mutually satisfactory agreement. Hence, MCC service providers should be aware of developing adequate agreements for mobile service provisioning. They need to balance the benefits of personalization against an acceptable level of privacy, since there is a fine line between practicality and intrusion.

2.2 Context services and quality of context

Adaptive services consume context information in order to adjust their services to the current configuration of users. They typically capture raw data from existing context sources, environmental sensors, or directly from the users themselves and their portable devices, and then they aggregate it and derive composite context information. High-level context information can also be obtained from a Context Management System (CMS). This managerial structure provisions the necessary infrastructure to collect, aggregate, and disseminate context information to various consumers, which require high-level context information for their operations. Raw context data originate from several sources wherein the subjects are mainly users, equipment, portable devices, and users' environment.

CMS structures have already been investigated in the literature as part of research efforts on context-aware services. Baldauf et al. [13] compared several CMSs along several dimensions such as sensing capability; context information representation and modeling; reasoning from and processing context information; maintaining historical data; and privacy and security considerations. Bettini et al. [14] described state of the art approaches for modeling context information and reasoning from it. These approaches contribute to the most popular methodologies that originate from modeling techniques used in database systems, and from ontology-based techniques used in knowledge representation. Perera et al. [15] surveyed a broad range of systems, techniques, methods, models, middleware solutions, and application models proposed by researchers to solve different issues in context-aware computing. They also identified some new trends pertaining to the importance of context awareness in the emerging IoT paradigm.

CMSs evolved into *Context Services* by embracing service-oriented conceptions in their implementation. Lei et al. [16] designed and implemented a middleware software to collect and disseminate context information in the form of a context service. They addressed extensibility issues of their context service to support heterogeneous primary context sources, privacy, and quality of context information (QoC). Schmidt et al. [17] implemented a generic service, which permits collection, discovery, and

monitoring operations of context information. These operations are exposed as Web services to facilitate their integration into heterogeneous environments. These Web services utilize Web Ontology Language (OWL) to deliver a semantic representation of context information whereby SPARQL endpoints are used to query and monitor context information.

With the proliferation and advances in sensing technologies, context services collect, process, and store significant amounts of context information. There is a growing trend toward deploying them on cloud infrastructures to offer cost-effective and more reliable context information within the service provider's premises. Cloud-based context services take advantages of the processing and storage capabilities of the cloud infrastructure. They can scale up when further computing and storage resources are needed due to an upsurge in demand for context information. Also, they can scale down once the demand for context information drops significantly. Li et al. [18] surveyed eleven context-aware middleware architectures. They compared these middleware architectures based on some technical parameters including architectural style, context abstraction, context awareness level, context reasoning, storage, service discovery, scalability, interoperability, fault tolerance, security and privacy, and cloud-based big-data analytics.

The heterogeneity of sensors used in various context sources raises concerns over the reliability of induced measurements and obtained raw data by adaptive applications that use the corresponding context-services. Also, the aggregation process of raw data can add additional biases. The concept of Quality-of-Context (QoC) has been proposed in several research studies [19, 20] to address these concerns. Krause and Hochstatter [20] defined QoC as:

Quality of Context (QoC) is any inherent information that describes context information and can be used to determine the worth of the information for a specific application. This includes information about the provisioning process the information has undergone ('history,' 'age'), but not estimations about future provisioning steps it might run through.

QoC might *impact* the quality of the services offered by adaptive services significantly. Indeed, low-quality context information might lead to making inappropriate decisions concerning the adaptation of the service. Several studies recognized the significance that QoC plays in the adaptation of services. Buchholz et al. [19] considered the following QoC indicators: *freshness*, *precision*, *spatial resolution*, *temporal resolution*, and the *probability of correctness*. Kim and Lee [21] added the following QoC indicators: *completeness*, *accuracy*, *representation consistency*, and *access security*. Neisse et al. [22] described a QoC framework, which relies on three QoC indicators: *up-to-dateness*, *precision*, and *resolution*. Hoyosa et al. [23] presented an extension to the context modeling language MLContext [24] for modeling QoC, where added features include constructs to express quality requirements, quality levels, and context situations (i.e., constraints on context information).

A *Context Level Agreement (CLA)* that represents the agreement between a context service provider and a context consumer has been suggested [25]. It describes common understandings and expectations of the two parties. QoC levels that the context provider should deliver are guaranteed throughout this agreement. The typical sections of CLA

are *Parties*, *Activation time*, *Scope*, *Context-level objectives* (CLOs), and *Penalties* [25].

3 Related work and motivation

There have been extensive works on personalizing Web experiences and cloud services. These works span a wide range of application environments and domain contexts. Next, we review some of the salient literature related to the investigation and evaluation of these recent developments.

3.1 Personalized content delivery services in smart devices

Mobile device standards experienced inherent mismatch in usability with desktop counterparts when receiving online information and services, over the last decade. The bulk of the incurred overhead has been arising from generating user interfaces to suit the technical properties of mobile devices. Subsequent standardization working groups released a set of specifications to tailor Web contents to mobile devices, such as the Mobile Web Initiative Device Description Working Group, back in 2008. These specifications consider a variety of mobile-device aspects such as screen size, provisioned markup-language and image format support, which extend the Web to mobile device capabilities. The issued guidelines by W3C (consortium [26, 27]), support the design of better and unified mobile-user experiences in order to decrease complexities stemming from the diversity of mobile devices. These efforts led to the definition of a repository of mobile device characteristics, which prompted Web content-providers and developers to adapt their content and delivery interfaces to corresponding mobile devices.

On the other hand, another standard specification maintained by the Open Mobile Alliance or OMA (formerly WAP Forum), provided an XML-structure to match user agents driving user preferences with device capabilities, and labelled User agent profile (UAProf). This standard depicts inherent mobile device attributes to support display preferences of Web users [28]. In doing so, the Composite Capabilities/Preferences Profile (or CC/PP) [29] outlines a schematic RDF structure describing this information that comprises components describing user preferences with respect to device properties in the form of attributes-value pairs that convey hardware characteristics such as screen size, image capabilities, manufacturer, etc., software features such as operating system version, audio/video encoders, etc., network bandwidth, as well as user preferences in terms of usability. Typically, a profile repository is maintained by corresponding mobile manufacturers in the form of a Web server that provides CC/PP profiles in response to HTTP requests.

3.2 Web and cloud services personalization

Web personalization has been the subject of several research efforts [30–33]. Online merchants and many organizations are increasingly using this alternative to deliver

customer value and achieve lucrative growth using sophisticated personalization tools; many of them are cloud-based [34]. The aim of Web personalization is to recommend Web pages that the user can visit or products or services that she/he can buy in e-commerce portal cases.

There are three common approaches for Web personalization [31], namely *utilization of rule-based systems*, *content-based filtering*, and *collaborative filtering*.

Rule-based personalization systems employ manually or automatically generated rules to recommend items to users [35, 36]. Numerous e-commerce Web portals use manual rule-based systems. For these portals, Web administrators specify personalization rules using users' profile, demographic, or domain specific information.

Content-based filtering systems generate a user profile based on the descriptions of items that the user had previously purchased or had expressed interest in doing so in the future. Web portals store these descriptions as a collection of features or items' characteristics associated with the user profile. Subsequently, these systems generate recommendations to the user by finding out similar items that match the user profile. Content-based recommender systems use item features to calculate items' similarities. Pasquale et al. [37] provided a review of the state of the art of content-based filtering systems used in several application domains, by describing both their classical and advanced techniques for representing items and user profiles. They also presented the most widely used techniques for learning user profiles.

Traditional collaborative filtering systems generate recommendations to the user based on information on a number of users who are sufficiently similar to the user [32, 38]. These systems involve matching the ratings of the users for purchased items with those of similar users to recommend to the user other items or products that he/she has not seen or rated before and might be interested in. Collaborative filtering represents the widely-used approach that several e-commerce Web portals employ to add intelligence to their systems, by capturing user profiles and interests, and recommend relevant content and items or services to their customers or visitors.

Some tradeoffs and limitations of the above traditional collaborative filtering approaches have been discussed in the literature [31, 32]. To cope with the limitations, several systems use *item-to-item collaborative filtering*, which is a variation of collaborative filtering approach [32, 38]. Rather than matching the user to similar customers, this approach matches the purchased and rated items to similar items, then combines those sufficiently similar items to generate a recommended list of items to the user. Many surveys [8–10] described and compared collaborative filtering algorithms. Yang et al. [39] surveyed collaborative filtering recommender systems that use information obtained from social networks.

Another approach for Web personalization relies on *Web usage mining* [40–43]. This approach is increasingly becoming popular among researchers and businesses alike, as it allows them to discover the users' browsing behavior. Data mining techniques employed in Web usage mining include clustering, association rules mining, and sequential patterns. Several research works investigated the effective impact of personalization on users' decision making [44–46]. Results showed that automated recommendations offered by online retailing were more effective than traditional recommendations made by human experts and other consumers.

With the growing adoption of cloud computing by businesses, personalization of cloud services is becoming a pressing issue. A number of frameworks have been proposed to cope with this issue. Gopalan et al. [47] proposed a cloud-based service architecture for personalized media recommendations. The user device side includes a thin client manager that runs the user interface, a client service manager that tracks the user activity and captures context, a local content manager that extracts metadata associated with the device, a context manager, and a communication manager. The server-side is deployed on the cloud and includes a recommendation engine manager, a user account manager, a context manager, and a pattern analyzer.

Guo et al. [48] proposed a framework to support the provision of personalized cloud services. Their proposed framework includes a client-side and a server-side component running on the cloud. The client-side manages the user profile and captures her/his activities. A user model is also built on the client-side and is updated on a regular basis. To adapt the service to the user preferences, the server-side fetches the user model from the client-side periodically and updates the user data accordingly.

Georgakopoulos et al. [49] presented the MediaWise project, which aims to expand existing media delivery systems with cloud-based personalization and collaboration capabilities. The proposed MediaWise Cloud platform aims to provide support for the creation, search, management, and consumption of multimedia content. In this work, the authors gave an overview of the platform's components, discussed the research objectives related to these components, and compared the features of existing content delivery networks against the envisioned objectives of the proposed platform.

Jang et al. [50] proposed CloudMat, a cloud-enabled context-aware personalization system of content fitness. CloudMat adapts the cloud service to the state of the user on the physical mat, and offers real-time feedback to the user. The cloud service content is adapted and displayed on an electroluminescent lighting layer on the mat to provide spatial guidance to the user. Sensors embedded in the mat capture the user's pose and timing activity, which are then disseminated to the cloud service. This service generates pose templates and metadata regarding the routine physical exercise to be delivered to the user. The utility of the proposed system has been shown through a case study on interactive yoga.

Hummer and Schulte [51] described an approach for context-aware personalization of mobile cloud services. The approach was illustrated using a connected cars scenario. Its three stages are aggregation of raw context events into high-level information, propagation of changes in context into user-specific service configurations, and integration with cloud services to perform context-triggered adaptations. Fan et al. [6] proposed an integrated personalization framework for SaaS-based cloud services. This framework employs a number of approaches including a client-side personalization, recommendation techniques, a dynamic cloud service composition engine, an ontology-based semantic integration, a user modeling technique, and some data-mining approaches.

3.3 Motivations and aims

As we mentioned in previous sections, most major Web services regularly personalize content and services for individual users based on their past service invocations.

Examples of these services include Amazon's personalized item-recommendations, Netflix's media recommendations, as well as Google's and Yahoo's personalized Web search results. With the proliferation of Internet-capable mobile handheld devices, users are increasingly requiring personalized content as it helps them to rapidly get relevant content from massive collections of data.

Our work aims at developing a framework to address the issue of service personalization in MCC environments. It aims at satisfying mobile user needs for provisioning personalized services. We assume that mobile cloud service providers compose these services from a set of in-house services and from third-party provided services using a given composition plan. The proposed framework integrates the user's profile, the profile of the mobile device in use, and the current context of the mobile user to adapt delivered personalized services.

The architecture proposed by Goapalan et al. [47] is to some extent close to ours as both of them consider client-side and server-side components. However, our framework focuses more on dynamic service composition from atomic services and service adaptation to deliver personalized services that meet the user's profile, the device profile, and the user context which can be obtained either from the user device or from external context-provider services.

The framework proposed by Guo et al. [48] differs from our proposed framework given that their user model is built and managed on the user device side. This means that the client machine should be powerful enough to compute the user model and handle the request from several servers, while fetching the user model. Our framework targets users with various devices mainly mobile ones with limited resources, and might obtain context information from third party providers of context services.

The framework proposed by Fan et al. [6] is the closest to our work as it shares nearly the same objectives as ours. However, there are few differences between the two frameworks. Our framework essentially targets mobile users who are effectively on the move and who might use various mobile devices that are characterized by limited resources. Their framework targets users with SaaS requests that essentially solicit powerful machines for rendering recommendation tasks. These tasks are however migrated from the server-side to the client-side to reduce the overhead on SaaS server and improve collecting user-profile data.

4 A framework for adaptive mobile service provisioning

4.1 Context

We are considering a service provisioning environment where mobile users access cloud services using minimal client resources on their devices. Storage and processing are all claimed from MCC provider as users rely on their devices anytime and anywhere to efficiently retrieve these services. Users do prefer flexible and personalized environments that take into consideration their preferences, with regard to functionality and expected quality-of-service (QoS) levels of the delivered services, as well as their current context. They might also use diverse mobile devices to access these services. So, the scope of this work takes into account the following considerations:

- User mobility.
- Heterogeneous mobile devices.
- User context.
- Tailored services.
- Personalized interfaces with similar look and feel on various devices.
- Cloud service composed from basic in-house and third party services.

We assume that the provisioned cloud service is the composition of two or more child services that can themselves aggregate several elementary or composite services. An elementary service is a basic service, which does not depend on other services. Child services might be developed in-house or obtained from third party providers. Service Level Agreements (SLAs) govern the provisioning process of these services in order to guarantee their levels of service as expected by the consumers of the composite service [52]. Violations of agreed upon SLAs may have an impact on the QoS offered to service-consumers. Numerous research works investigated the resulting QoS of composite services by considering various composition patterns (sequential, parallel, XOR, AND) [53, 54].

4.2 System architecture

In this section, we propose a framework for implementing an adaptive system that supports mobile users connecting to the cloud service provider platform using various mobile devices. The main tasks of the cloud service provider are:

- Managing the advertised services and their composition plan.
- Managing the acquisition of context information from potential context service providers.
- Providing a personalized service to mobile users based on their device model, service preferences, and context of use.
- Adapting and packaging external services that are exported to third party providers to facilitate their conversion and adaptation to the required format expected by the third-party provider user and device profiles.

The overall architecture of our proposed framework is illustrated in Fig. 1. The architecture includes two parts. The first part contains the components that run on the mobile device side. The second part concerns the personalized cloud service, which includes in its core: the adaptation and personalization engine, the service composition manager, and the context acquisition manager. The details of the above parts are further elaborated next.

4.3 User device components

Figure 2 depicts the components of the personalization service on the mobile user device and comprises the following elements:

- (a) *User Profile Manager (UPM)* manages a local copy of the user profile (preferences and user's model), on the user's device. Its main tasks are:

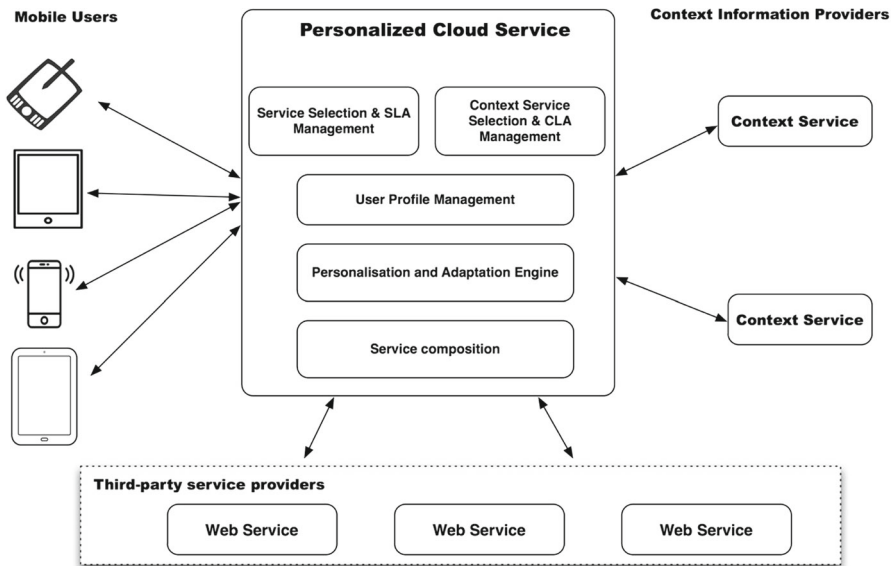


Fig. 1 Architecture for personalized cloud service provisioning for mobile users

- Providing user information (profile data, identification).
 - Managing and synchronizing the user profile with the cloud service.
 - Providing a local personalization of the service offering. In doing so, UPM adapts the display of the service according to the user's preferences and device capabilities.
- (b) *User Service Access Manager (USAM)* manages the access and the interactions with the cloud service. *USAM* ensures the necessary adaptation and conversion in collaboration with *UPM* and *USIM*. It synchronizes the service content with the cloud service.
- (c) *User Service Interface Manager (USIM)* is in charge of maintaining the profile of the user's device and ensures that the display of service content is adjusted to the user preferences and device capabilities. This component provides the interface for the user to login and optionally input his initial preferences and feedback.
- (d) *User Local Context Manager (ULCM)* identifies and retrieves the mobile user activity context. It aggregates context data obtained from different sources such as GPS, other sensors of the mobile device, and other applications such as calendar. Then, it sends an aggregated context information to the Context Acquisition Manager on the server side via *USAM*.

4.4 Cloud service-side components

As depicted in Fig. 3, the cloud service has the following components: *Profile Manager (PM)*, *Device Profile Manager (DPM)*, *Service Delivery Manager (SDM)*, *Adaptation Engine*, *Service Offerings Manager (SOM)*, *Service Selection Manager (SSM)*, *SLA Manager (SLAM)*, *Service Composer (SCOM)*, *Context Selection Manager (CSM)*, *CLA Manager (CLAM)*, and *Policy Manager (PolM)*.

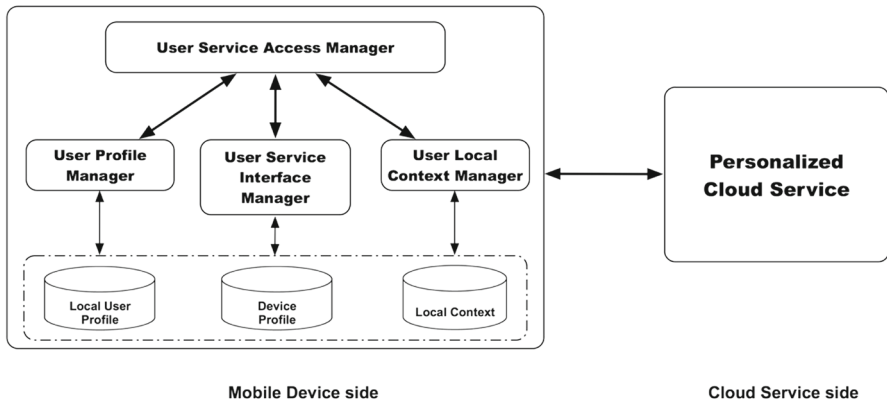


Fig. 2 Personalization service components on the mobile user device

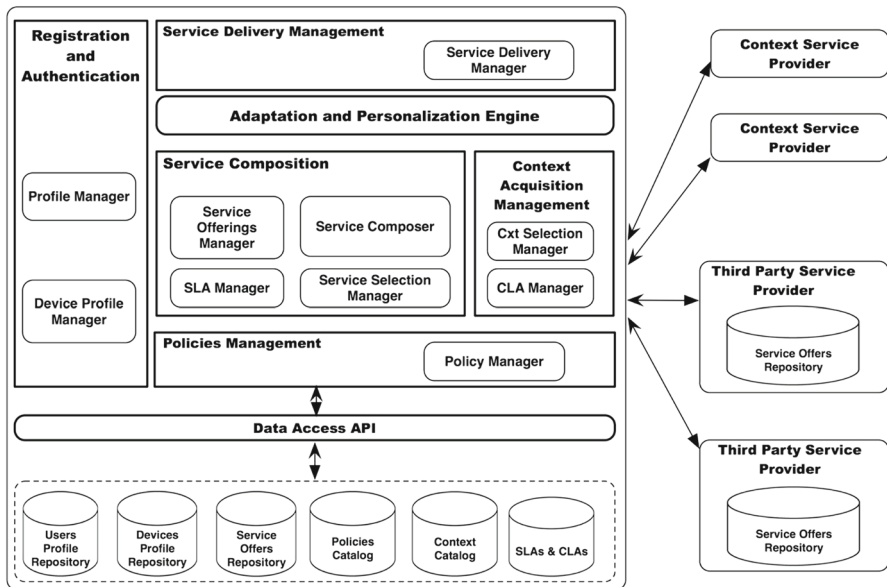


Fig. 3 Cloud service architecture for personalized and adaptive service provisioning

(a) The *Profile Manager (PM)* manages information about users. Its main tasks are:

- Authenticating users.
- Registering each new user into the users' Profile Repository.
- Managing and ensuring the consistence of the database containing users' profiles.
- Giving access to user profiles data.
- Initiating UPM and USIM on the remote device.
- Checking the versions of UPM and USIM that reside on remote devices and automatically downloading any necessary updates.

- (b) *Device Profile Manager (DPM)* manages information about various devices. Its main tasks are:
 - Registering new devices of the user into the Devices Profile Repository.
 - Managing and ensuring the consistency of the database containing devices' profiles.
- (c) *Service Delivery Manager (SDM)* manages information about service offerings and their delivery plans. Its main tasks are:
 - Providing an interface for defining service offerings and related information (composition plan).
 - Receiving service requests from devices and giving access to services.
 - Working with the adaption engine to adapt a service composition plan based on user profile, preferences, and context.
 - Packaging services to meet user and device profiles.
 - Initiating USAM and USIM on a mobile user's device.
- (d) *Personalization and Adaptation Engine* is in charge of adapting composed services according to the mobile user profile, preferences, and context. In Sect. 6, we will describe the process for obtaining context information from context services and the process for selecting appropriate services that make up the composite services. The challenge at this level is the heterogeneity in both service offerings and context services' offerings, which are usually described using various metadata models.
- (e) *Service Offerings Manager (SOM)* is in charge of managing in-house service offerings and accessing third-party ones.
- (f) *Service Selection Manager (SSM)* implements policies for the selection of suitable services that can be used in the composition process, based on the requirements of the cloud service provider which houses the services and the actual service providers' QoS requirements.
- (g) *SLA Manager (SLAM)* is in charge of carrying out the SLA negotiation process with service providers following their selection by the Service Selection Manager.
- (h) *Service Composition Manager (SCOM)* composes the cloud service from the selected services by the Service Selection Manager using the composition plan.
- (i) *Context Selection Manager (CSM)* implements diverse algorithms for the selection of appropriate context providers based on QoC requirements and the context providers' QoC offerings.
- (j) *CLA Manager (CLAM)* is in charge of carrying out the CLA negotiation process with the context provider that the Context Selection Manager selected.
- (k) *Policy Manager (PolM)* is in charge of defining and managing various policies such as authentication and authorization policies.

The back-end databases of the framework include:

- Users' Profile Repository: for each user, the system maintains a profile that has two components, the user's model and the user preferences regarding the service offerings, interfaces and content display.
- Device Profiles Repository: contains a description of each device including its features and capabilities that are useful for the service provision (screen size, bandwidth

limit, color schemes, resolution, etc.). Some features that the system can automatically detect (operating system, browser, plug-ins) are not stored in the repository but integrated to the profile when initializing USIM.

- **Service Offerings Repository:** is the service registry, which contains service offerings. For each offering, the system maintains its composition plan.
- **Context Catalog:** contains context information about the mobile users who are using the cloud service. Context information is directly obtained from the users' mobile devices and from various context services offered by third-party providers.
- **Policies Catalog:** contains the policies of the various services used in the composition of the cloud service and the context service from which context information is obtained.
- **SLAs and CLAs Repository:** contains the agreed-upon SLAs and CLAs with third party service providers and context services respectively. It also contains SLA and CLA templates used during the negotiation process with these providers.

5 Adaptation structures for personalized service delivery

5.1 Services registry

The cloud service offering relies on a set of basic or composite services for its service delivery to mobile users. Each child service is either an in-house built service or imported from an external provider. The services registry defines the relationships between these child services. The relationships consist of prerequisite, similarity, and substitute relationships.

5.2 Composition plan

The composition plan defines the sequencing of the service offerings and some time constraints or deadlines by which they are expected to be executed to ensure service delivery to users. Different approaches have been considered for planning the composition of services. Digiamietri et al. [55] provided a review of current planning approaches and proposed a solution based on artificial intelligence planning for automatic composition of services.

5.3 User model

The mobile user profile represents a characterization of the user at a specific moment in time. Data in the profile is stored based on explicit information given by the user or inferred by the system from log files, for example, that trace previous usage of the service.

Information about the mobile user is collected in the user model, which describes the features of the user. It is the basis for service personalization. Using the user model, the mobile service can differentiate between different users and adapt the service provisioning and interfaces to specific user needs. A perfect user model would include

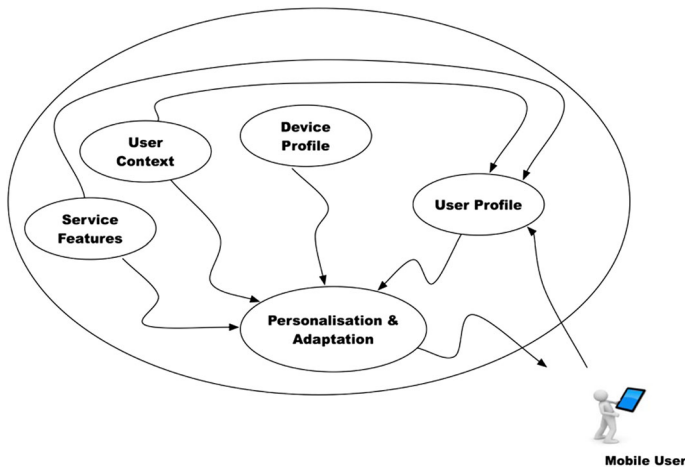


Fig. 4 User model for service adaptation and personalization

all features of the user behavior and knowledge on service utilization and appreciation of service performance. As the construction of the user model is challenging, many research works used simplified models.

Three main aspects that need to be considered in designing the user model include: information about the user that need to be included in the model and the process for obtaining such information, its representation, and the process of creating and updating the model [56]. Management of the user model involves the following activities: deciding on the user data to include in the model, representing that data, and continually updating the model using inferred data from the interactions of the mobile user with the cloud service.

As we are coping with a mobile service provisioning system, the main information to store in the user model is the knowledge of the user about the service being consumed. This information is repetitively collected during the interaction of the user with the service. It is also used for updating the user model (see Fig. 4).

To model users, we consider a fuzzy overlay representation over the service concepts and features, as introduced in a previous work [57]. Each concept in the model is associated to a fuzzy value that represents the assessment of the user's knowledge regarding that concept.

The system uses two different versions of the user model: a global user model and a local user model. The global user model is stored in the User Profile Repository and represents the overall concepts reported to the system about the user or accessed from other external service offerings. In addition to the service concepts and associated fuzzy values, this model also represents the relationships between concepts (prerequisite, similarity, and substitute). The local user's model is managed by the User Profile Manager within the mobile user's device and is related to a specific service offering. This model represents only the service concepts and associated values. It is refined based on user's interactions with the service provider when accessing the services, and used to update the global user's model. The local user model is initialized from the global user model.

5.4 Service provisioning strategy

The service provisioning strategy is a set of rules that implement the adaptation controls for the service offerings. It consists of rules for sequencing service offerings, rules for adding or dropping service offerings and rules for selecting between similar or equivalent service offerings. Next section describes an algorithm for the selection of potential context services. A similar process can be adapted to the selection of services that can be used in the composition, by using QoS (quality of service) attributes instead of QoC attributes.

6 Context service selection engine

In this section, we describe the process of selecting potential context services that can provide up-to-date contextual information about the user and her/his environment based on the cloud service provider's QoC requirements. Service selection in general has been largely investigated in the area of service oriented architecture and Web services using different approaches and algorithms. These approaches include, but are not limited to:

- Agent-based approach.
- QoS ontology-based approach.
- Muti-attributes decision making approach.
- Particle Swarm Optimization based approach.
- Broker based approach and maximization of application-specific utility function.
- Fuzzy logic based approach.

In the following, we describe an algorithm capable of evaluating potential context service offers by taking into account QoC requirements of the cloud service provider and the various QoC service offerings.

Let $Q = \{Q_1, Q_2, \dots, Q_n\}$ be the list of QoC attributes, such as freshness of context information, and its probability of correctness. These QoC attributes may be classified into two categories. In the first category, we find *QoC benefit attributes*, such as probability of correctness, which the cloud service provider wants to maximize. In the second category, we find *QoC cost attributes*, such as freshness of context information that the cloud service provider wants to minimize.

Let $CS = \{CS_1, CS_2, \dots, CS_K\}$ be the list of potential Context Services, which are capable of providing the type of context information requested by the cloud service provider. Let $R = \{R_1, R_2, \dots, R_n\}$, be the vector of quality requirements of the cloud service provider for each QoC attribute in Q .

The following vector expresses the QoC offering of the Context service CS_m :

$$Q^m = \{Q_1^m, Q_2^m, \dots, Q_n^m\}$$

The above QoC attributes are expressed in different units and cover a range of values. To be able to sort and rank the context services' offerings with regard to their QoC offerings, we need to normalize values of QoC offers and define utility functions

that will allow mapping the vector of QoC values into a single real value. We define the maximum and minimum values of QoC attribute Q_i considering all the offers of the potential context services in CS, namely:

$$Q_i^{\max} = \max_{1 \leq j \leq K} (Q_i^j)$$

$$Q_i^{\min} = \min_{1 \leq j \leq K} (Q_i^j)$$

The requirement of the cloud service provider R_i for Q_i may fall in the range $[Q_i^{\min}, Q_i^{\max}]$ or outside of this range. To take into account the requirements of the cloud service provider during the normalization, we define the two following values:

$$Q_i^{\max} = \max(R_i, Q_i^{\max})$$

$$Q_i^{\min} = \min(R_i, Q_i^{\min})$$

We define q_i^r , the normalized value for the context service CS_m offer with regard to the QoC attribute Q_i in Eq. (1) as:

$$q_i^m = \begin{cases} \frac{Q_i^{\max} - Q_i^m}{Q_i^{\max} - Q_i^{\min}} \text{ for cost QoC attributes} \\ \frac{Q_i^m - Q_i^{\min}}{Q_i^{\max} - Q_i^{\min}} \text{ for benefit QoC attributes} \end{cases} \quad (1)$$

Let $RN = \{r_1, r_2, \dots, r_n\}$, with $0 \leq r_i \leq 1$, be the vector of normalized values of the quality requirements of the cloud service provider.

We define the utility $u_{i,m}$ as the level of satisfaction of the offering of context service CS_m with respect to the cloud service provider requirement for the quality attribute Q_i .

$$u_{i,m} = \frac{q_i^m}{r_i} \quad (2)$$

When $u_{i,m}$ is greater than 1, it means that the offer of context service CS_m exceeds the expectation of the cloud service provider with regard to quality attribute Q_i . When it is smaller than 1, it means that the offer does not meet its expectation for Q_i .

The cloud service provider may assign an *importance weight* to each quality attribute. Let $W = \{w_1, w_2, \dots, w_n\}$ be the weight vector associated with the quality attributes respectively. We define a Utility function (weighted combined level of satisfaction) associated with the offered context service CS_m as:

$$U_m = \sum_{i=1}^n w_i u_{i,m} \quad (3)$$

Equation (3) allows ranking QoC offers of potential context services that are capable of offering the context information required by the cloud service provider. The best context service offer corresponds to the offer that corresponds to the highest value of the utility function U_m . Algorithm 1 summarizes the steps of the algorithm.

A similar algorithm, often referenced in literature, for comparing services based on the user QoS requirements is the TOPSIS algorithm developed by Hwang and Yoon [58]. The TOPSIS method aims to determine the best alternative from the concepts of the compromise solution. The solution that has farthest Euclidean distance from the negative ideal solution and the the shortest Euclidean distance from the ideal solution corresponds to the best compromise solution. The algorithm has six main steps [59]:

1. Normalization of the decision matrix.
2. Calculation of weighted normalized decision matrix.
3. Determination of the positive and the negative ideal solutions (PIS and NIS).
4. Calculation of the separation measures for each alternative from the PIS and the NIS.
5. Calculation of the relative closeness coefficient to the positive ideal solution.
6. Ranking the alternatives.

Section 8 describes a case scenario in which we apply our proposed algorithm and the TOPSIS algorithm to rank the offers of a numbers of context services.

Algorithm 1. Context services selection algorithm

```

Input:
 $Q = \{Q_1, Q_2, \dots, Q_n\}$  //quality-of-context indicators
 $RN = \{r_1, r_2, \dots, r_j, \dots, r_n\}$  // QoC Requirements of the cloud service
                                     // provider expressed as normalized
                                     // values using Eq. (1)
 $W = \{w_1, w_2, \dots, w_j, \dots, w_n\}$  //Cloud service provider weight vector
 $Q^m = \{Q_1^m, Q_2^m, \dots, Q_n^m\}$  // QoC offering of context service  $CS_m$ 
                                     //  $1 \leq m \leq K$ 

Output
 $U = \{U_1, U_2, \dots, U_i, \dots, U_n\}$  //Global utility vector
// Select Context Service offering the highest Utility

For each offer  $m$ ,  $1 \leq m \leq K$ 
    //initialize global utility of offer  $m$  to zero
     $U_m \leftarrow 0$ 
    For each  $q_i \in Q$ 
        // Compute the normalized value using Eq. (1)
         $q_i^m \leftarrow$  normalized value of offer  $m$  for  $q_i$ 
        // Compute utility functions for  $q_i$ 
         $u_{i,m} \leftarrow q_i^m / r_i$ 
         $U_m \leftarrow U_m + w_i u_{i,m}$ 
    End For each
End For each
Select  $CS_j$  with  $U_j = \max(U_m)$  for  $1 \leq m \leq K$ 

```

7 Integration and adaptation of third party service offerings

In modern enterprise systems, applications are built from smaller services, described by service offering (SO) descriptions. SOs allow customizing services for each individual

user or even for a whole organization. A single service offering may be used in diverse contexts for multiple purposes. In SO repositories, each SO has descriptive metadata allowing it to be easily found in a search process and integrated with other SOs to build much larger services. SOs metadata can be written in XML or in any other proprietary format. The OASIS WSDL Standard for Service Offering Metadata is an internationally recognized standard for describing services. Also, the *Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)* [60] provides an application-independent interoperability framework based on metadata harvesting. A harvester is a client application that issues OAI-PMH requests. It is mainly operated by a service provider as a means of collecting metadata from data repositories.

As the goal of the cloud service provider is to offer an adaptive service, it should distinguish important contextual information for its business. The adapted content to the service consumer's context may be for instance presenting information on new items, articles, promotions, and so on. As we mentioned earlier, context information may be obtained directly from the user's mobile device, and/or from third party services that we referred to as context information services (CIOs).

With the heterogeneity in context offering descriptions and context information modeling, it is necessary to translate these offerings into a common model used by the adaptation engine [61]. ContextML [62], a light weight XML-based context representation schema, is a potential candidate for that common model. In ContextML, context information is categorized into scopes and may be related to different types of subjects, such as user and device. It may be incremented with attributes to describe various QoC indicators.

The proposed framework is open to third-party providers that can expose their service offerings as Web services. Their services can be plugged into the proposed framework. With the increasing support to the concept of open service offerings' repositories (SORs), it will be possible to search for appropriate services that can be integrated with local services based on a mobile user's profile and a composition plan.

The heterogeneity of the metadata used to describe SOs requires a process for adapting the service composition, built from various in-house and imported SOs, to the user's profile, device profile, and context. The framework uses a common metadata model to describe SOs. This process, depicted in Fig. 5, involves the following steps:

- *Step 1* The process starts by identifying the basic services, which are required for the cloud service composition, and the context information required for the service adaptation based on the user profile. The identified basic services need to be acquired from third-party service providers if they are not offered in-house. The *Service Selection Engine* selects appropriate SOs based on the QoS requirements of the cloud service composition. Likewise, the *Context Service Selection Engine* selects appropriate CIOs, able to offer required context information, based on the QoC requirements of the cloud service composition.
- *Step 2* The *SO Description Adapter* translates the metadata of the SOs, selected in previous step, into a common metadata model for describing basic service offerings. In a similar way, the *CIO Description Adapter* translates the metadata of the CIOs, also selected in previous step, into a common metadata model for describing context information.

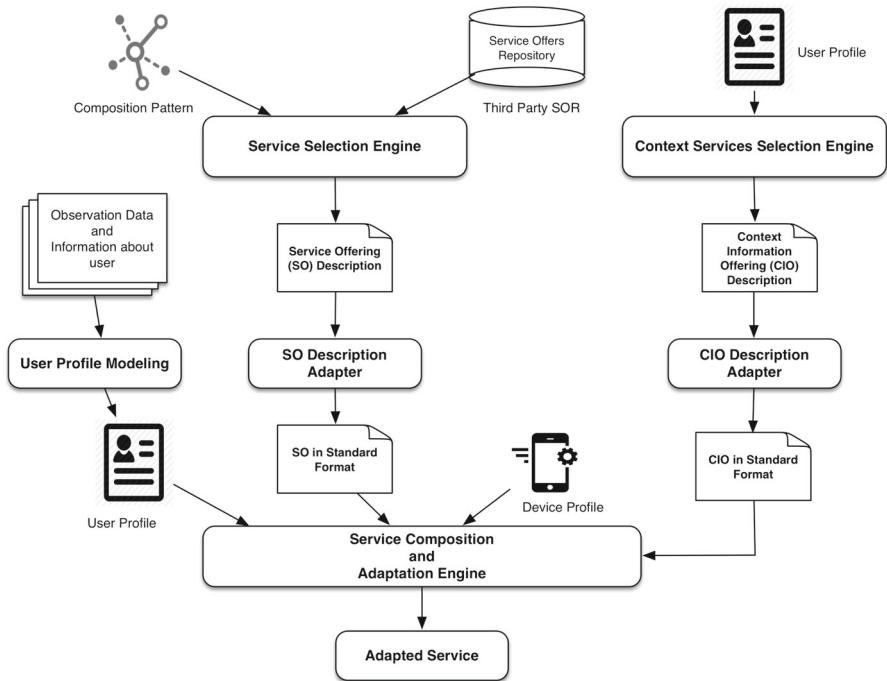


Fig. 5 Context-aware adaptation of third party service offerings

- **Step 3** The resulting metadata of selected SOs and CIOs, obtained in the previous step, the user profile, and the device profile are the input of the *Service Composition and Adaptation Engine*. This later generates the composed service, which adapts the service offering to the user profile, device profile, and current context of the user.

8 Application and performance analysis

In this section, we consider an application scenario in which the required context information is the *temperature* reading at the mobile user's location. Many context service offerings are available to provide that information each one with its own QoC offerings. This application scenario is used to illustrate the process of context services selection.

We assume that the QoC requirement of the cloud service are: *Freshness*=6 ms, *Precision*=99.96%, and *Probability of Correctness*=95.96%. Table 1 shows the QoC values offered by 24 hypothetical context service providers. Table 2 depicts the normalized values of each of these QoC attributes. Table 3 lists the weights that the MCC service provider assigns to these QoC parameters according to the mobile user requirements. By using our selection algorithm, normalized values are calculated for each QoC parameter while differentiating between benefit-driven QoC attributes and

Table 1 QoC offerings of hypothetical context service providers

Cxt_ID	Freshness (ms)	Precision	Probability of correctness
1	10	0.98955	0.95954
2	8	0.97953	0.98953
3	7	0.99962	0.97962
4	9	0.98864	0.99886
5	6	0.97954	0.93974
6	7	0.96858	0.96312
7	6	0.97771	0.91547
8	8	0.98675	0.93457
9	7	0.99879	0.92245
10	8	0.97256	0.98698
11	7	0.97671	0.95326
12	6	0.98976	0.94785
13	5	0.95787	0.90235
14	9	0.98983	0.96874
15	8	0.98288	0.97965
16	8	0.99453	0.95548
17	7	0.96967	0.93658
18	7	0.98562	0.92896
19	9	0.96875	0.97895
20	8	0.97772	0.98456
21	9	0.96857	0.99124
22	7	0.95677	0.93659
23	6	0.98952	0.94857
24	8	0.97692	0.98963

cost-driven QoC attributes. Then, using the weight table, the aggregate utility function is calculated for each context service offering.

Table 4 depicts the ranking table generated by MCC service using our proposed algorithm. The offer of the Cxt provider with ID 3 is the best offer as it maximizes the aggregated utility. In parallel with that, we also use the TOPSIS method [58] for ranking the Cxt offerings. First, we identify the positive ideal solution (PIS) and the negative ideal solution (NIS). Then, we calculate for each Cxt offering the separation from PIS and from NIS. Afterwards, we calculate the closeness to the ideal solution (CIS). Table 5 depicts the ranking using the TOPSIS method.

To our surprise, the above results show that both algorithms have obtained the same five best Cxt providers (Cxt ID: 3 is the best one). Subsequent Cxt offerings in the two rankings are very close, especially for the first ten best offerings. This process results in identifying a suitable Cxt provider for temperature provisioning by considering the three QoC attributes: freshness, precision, and probability of correctness.

Table 2 Normalized QoC offerings of context service providers

Cxt_ID	N. freshness	N. precision	N. probability of correctness
1	0.00	0.83	0.59
2	0.40	0.66	0.90
3	0.60	1.00	0.80
4	0.20	0.82	1.00
5	0.80	0.66	0.39
6	0.60	0.48	0.63
7	0.80	0.63	0.14
8	0.40	0.78	0.33
9	0.60	0.99	0.21
10	0.40	0.55	0.88
11	0.60	0.62	0.53
12	0.80	0.83	0.47
13	1.00	0.30	0.00
14	0.20	0.84	0.69
15	0.40	0.72	0.80
16	0.40	0.91	0.55
17	0.60	0.50	0.35
18	0.60	0.77	0.28
19	0.20	0.48	0.79
20	0.40	0.63	0.85
21	0.20	0.48	0.92
22	0.60	0.28	0.35
23	0.80	0.83	0.48
24	0.40	0.62	0.90

Table 3 Weighted table of QoC attributes

	Freshness (%)	Precision (%)	Probability of correctness
Weight	0.40	0.35	0.25

9 Conclusion

The phenomenal proliferation of Internet-enabled mobile devices, such as tablets and smartphones, having multimedia and sensing capabilities, has triggered over the past few years the development of new applications and services of all kinds. These services include shopping-on-the-go, online hotel booking, accessing educational resources, and many more. However, these devices are still suffering from what is known as resource (computing and storage) poverty. Mobile cloud computing is a new paradigm

Table 4 Ranking of Cxt offerings using our proposed algorithm

Cxt_ID	Utility (%)	Rank
3	79.02	1
23	73.04	2
12	73.00	3
5	64.90	4
9	63.72	5
2	61.79	6
16	61.77	7
4	61.55	8
15	61.20	9
24	60.28	10
20	59.44	11
11	58.74	12
18	57.67	13
7	57.54	14
10	57.04	15
6	56.52	16
14	54.45	17
8	51.79	18
13	50.49	19
17	50.28	20
21	47.80	21
19	44.72	22
1	43.90	23
22	42.71	24

that aims at addressing this shortcoming. Several research efforts studied the opportunities and challenges of this paradigm. Personalization of services, in the context of mobile cloud computing, remains a challenging issue due to the variety of mobile devices in the market and the heterogeneity across users' profiles and preferences. The sensing capabilities of these devices add a new dimension, which is the user context, to the personalization challenge.

In this work, we proposed a conceptual framework that could serve as the basis for developing cloud services that can deliver personalized services. We described the different components to be deployed on the service provider side and the components to deploy on the mobile user's device. The mobile cloud service provider typically composes its service from a set of in-house basic services and from third party services using a composition plan. We described how to achieve the adaptation of services by taking into account the user's profile, the device profile, and the user and her/his environment context. The information structures used to support personalized service delivery include a service registry, the composition plan, the user model, and a service provisioning strategy. We also proposed an algorithm for the effective selection of suitable context services that meet the cloud service provider requirements in terms

Table 5 Ranking of Cxt offerings using the TOPSIS method [58]

Cxt_ID	CIS (%)	Rank
3	72.00	1
23	71.70	2
12	71.59	3
5	64.37	4
9	59.97	5
7	57.82	6
11	56.87	7
18	56.25	8
16	56.03	9
15	55.01	10
2	54.99	11
6	54.15	12
24	53.73	13
13	53.38	14
20	53.28	15
4	51.72	16
10	51.08	17
17	49.40	18
8	48.52	19
14	47.67	20
22	43.01	21
21	41.95	22
1	40.10	23
19	39.27	24

of context information and quality of context. Finally, we evaluated our framework in the context of an application scenario to illustrate the performance of the proposed algorithm. Our future work will further investigate a wide range of application contexts which integrate Internet of Things (IoT) services that potentially unleash further context patterns.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Dinh HT, Lee C, Niyato D, Wang P (2013) A survey of mobile cloud computing: architecture, applications, and approaches. *Wirel Commun Mob Comput* 13(18):1587–1611
2. Xu Y, Mao S (2013) A survey of mobile cloud computing for rich media applications. *IEEE Wirel Commun Mag* 20(3):46–53

3. Riecken D (guest ed.) (2000) Personalized views of personalization (special section). *Commun ACM* 43:8
4. Hagen P, Manning H, Souza R (1999) Smart personalization. Forrester Research, Cambridge, MA
5. Dew MAA, Achmad Nizar H, Muhammad Rifki S, Yu-Qian Z (2017) Trust transfer and its effects on the continuance usage of mobile service in B2C E-marketplaces. In: Pacific-Asia conference on information systems
6. Fan H, Khadeer HF, Younas H, Hussain OK (2015) An integrated personalization framework for SaaS-based cloud services. *Future Gener Comput Syst* 53:157–173
7. Liao CL, Lee SJ (2016) A clustering based approach to improving the efficiency of collaborative filtering recommendation. *Electron Commerce Res Appl* 18:1–9
8. Cacheda F, Carneiro V, Fernandez D, Formoso V (2011) Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high performance recommender systems. *ACM Trans Web* 5(1):2:1–2:33
9. Ekstrand MD, Riedl JT, Joseph AK (2011) Collaborative filtering recommender systems. *Found Trends Hum Comput Interact* 4(2):81–173. <https://doi.org/10.1561/1100000009>
10. Hameed MA, Jadaan OA, Ramachandram S (2012) Collaborative filtering based recommendation system: a survey. *Int J Comput Sci Eng* 4(5):859–876. <http://connection.ebscohost.com/c/articles/82397440/collaborative-filtering-based-recommendation-system-survey>
11. Badidi E, Routaib H (2015) A conceptual framework for personalization of mobile cloud services. In: The international conference on cloud computing technologies and applications—CLOUDTECH 2015, 2–4 June 2015
12. Zahi J (2010) Personalized web services for web information extraction. *Int J Web Serv Pract* 5(1):22–31
13. Baldauf M, Dustdar S, Rosenberg F (2007) A survey on context-aware systems. *Int J Ad Hoc Ubiquitous Comput* 2(4):263–277
14. Bettini C, Brdiczka O, Henriksen K, Indulska J, Nicklas D, Ranganathan A, Riboni D (2010) A survey of context modelling and reasoning techniques. *Pervasive Mob Comput* 6(2):161–180
15. Perera C, Zaslavsky AB, Christen P, Georgakopoulos D (2014) Context aware computing for the internet of things—a survey. *IEEE Commun Surv Tutor* 16(1):414–454
16. Lei H, Sow DM, Davis JS, Banavar G, Ebling MR (2002) The design and applications of a context service. *Mob Comput Commun Rev* 6(4):45–55
17. Schmidt H, Flerlage F, Hauck FJ (2004) A generic context service for ubiquitous environments. In: Proceedings of the IEEE international conference on pervasive computing and communications (PERCOM), pp 1–6
18. Li X, Eckert Martínez JF, Rubio G (2015) Context aware middleware architectures: survey and challenges. *Sensors* 15(8):20570–20607
19. Buchholz T, Kpper A, Schiffrers M (2003) Quality of context: what it is and why we need it?. In: Proceedings of the 10th international workshop of the HPOpenView University association (HPOVUA)
20. Krause M, Hochstatter I (2005) Challenges in modelling and using quality of context (QoC). In: Mobility aware technologies and applications. Springer, Berlin, Heidelberg, pp 324–333
21. Kim Y, Lee K (2006) A quality measurement method of context information in ubiquitous environments. In: ICHIT'06. Washington
22. Neisse R, Wegdam M, Van Sindren M (2008) Trustworthiness and quality of context information. In: Proceedings of the 9th international conference for young computer scientists (ICYCS'08), pp 1925–1931
23. Hoyos JR, García-Molina J, Botía JA, Preuveneers D (2016) A model-driven approach for quality of context in pervasive systems. *Comput Electri Eng* 55:39–58
24. Hoyos JR, García-Molina J, Botía JA (2013) A domain-specific language for context modeling in context-aware systems. *J Syst Softw* 86(11):2890–2905. <https://doi.org/10.1016/j.jss.2013.07.008>
25. Badidi E (2014) A collaborative framework for QoC-driven context information provision. *J Internet Technol Secur Trans (JITST)* 3(3):329–337
26. W3C (2008) Mobile web best practices 1.0, basic guidelines. W3C recommendation, 29 July 2008. <http://www.w3.org/TR/2008/REC-mobile-bp-20080729/>
27. W3C (2010) Mobile web application best practices. W3C recommendation 14 Dec 2010. <http://www.w3.org/TR/2010/REC-mwabp-20101214/>
28. Open Mobile Alliance (2001) WAG UAProf, version 20 Oct 2001. <http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf>

29. W3C (2007) Composite capability/preference profiles (CC/PP): structure and vocabularies 2.0. W3C working draft, 30 April 2007. <http://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430/>
30. Subramaniam S (2014) Frontiers in E-commerce personalization. In: The 20th ACM international conference on knowledge discovery and data mining, SIGKDD
31. Anil NK, Kurian SB, Varghese SM (2013) Multidimensional user data model for web personalization. *Int J Comput Appl* 69(12):32–37
32. Köhler S, Wöhner T, Peters R (2016) The impact of consumer preferences on the accuracy of collaborative filtering recommender systems. *Electron Markets* 26:369–379
33. Moawad IF, Talha H, Hosny E, Hashim M (2012) Agent-based web search personalization approach using dynamic user profile. *Egypt Inform J* 13(3):191–198
34. Website Magazine (2013) 20 wonderful web personalization tools for merchants. <http://www.website-magazine.com/content/blogs/posts/pages/20-wonderful-web-personalization-tools-for-merchants.aspx>
35. Paranjape VP, Deshpande U (2013) A stock market portfolio recommender system based on association rule mining. *Appl Soft Comput J* 13(2):1055–1063
36. Kim YS, Yum BJ (2011) Recommender system based on click stream data using association rule mining. *Expert Syst Appl* 38(10):13320–13327
37. Pasquale L, Gemmis DM, Semeraro G (2013) Content-based recommender systems: state of the art and trends. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) *Recommender systems handbook*. Springer, Berlin. <https://doi.org/10.1007/978-0-387-85820-3>
38. Wang J, Yin J (2013) Combining user-based and item-based collaborative filtering techniques to improve recommendation diversity. In: *Proceedings of the 6th international conference on biomedical engineering and informatics*, Hangzhou, China
39. Yang X, Guo Y, Liu Y, Steck H (2014) A survey of collaborative filtering based social recommender systems. *Comput Commun* 41:1–10
40. He D, Wu P, Wang J (2015) Personalized recommendation of E-commerce website category hierarchy based on web usage mining and multidimensional scaling. In: *IEEE 12th international conference on e-business engineering*, pp 80–86
41. Dhanamma J (2013) Web usage mining: pattern discovery and forecasting. *Int J Data Wareh Min* 2(4):187–190
42. Dewgun TK, Chauhan PS (2015) A survey on web usage mining: process, techniques and applications. *Int J Eng Res* 4(4)
43. Lopes P, Roy B (2015) Dynamic recommendation system using web usage mining for E-commerce users. *Procedia Comput Sci* 45:60–69
44. Li Y, Wang Y (2013) Social influence from personalized recommendations to trusting beliefs of websites: intermediate role of social presence. In: Kotzé P, Marsden G, Lindgaard G, Wesson J, Winckler M (eds) *Human–computer interaction—INTERACT 2013. Lecture notes in computer science*, vol 8119. Springer, Berlin
45. Hofmann J (2016) Analyzing the influence of product features and recommendations on buying decisions: a new realistic experimental web shop and analysis environment. *Dissertation Universität Duisburg-Essen*
46. Faridani V (2017) Collaborative filtering-based recommender systems by effective trust. *Int J Data Sci Anal* 3:297–307
47. Gopalan KS, Nathan S, Bhanu Teja CH, Channa AB, Saraf P, Shanker G (2011) Cloud based service architecture for personalized media recommendations. In: *Fifth international conference on next generation mobile applications and services*, pp 19–24
48. Guo H, Chen J, Wu W, Wang W (2009) Personalization as a service: the architecture and a case study. In: *Proceedings of the first international workshop on cloud data management, CloudDB'09*, ACM, New York, NY, pp 1–8
49. Georgakopoulos D, Ranjan R, Mitra K, Zhou X (2012) MediaWise—designing a smart media cloud. In: *The proceedings of the international conference on advances in cloud computing (ACC-2012)*, 26–28 July 2012. Bangalore
50. Jang KJ, Ryo J, Telhan O, Mangharam R (2015) Cloud Mat: Context-aware personalization of fitness content. In: *IEEE international conference on services computing (SCC 2015)*, pp 301–308. <https://doi.org/10.1109/scc.2015.49>
51. Hummer W, Schulte S (2015) Context-aware personalization for smart mobile cloud services. *ICSOC Workshops* 9586(14):171–183

52. DiModica G, Tomarchio O, Vita L (2009) Dynamic SLAs management in service oriented environments. *J Syst Softw* 82(5):759–771
53. Li H, Wang Y, Yuan J (2014) The calculation of QoS aggregation functions for composite web services. *J Netw* 9(9):620–628
54. Sheng QZ, Qiao X, Vasilakos AV, Szabo C, Bourne S, Xu X (2014) Web services composition: a decade's overview. *Inf Sci* 280:218–238
55. Digiampietri LA, Perez-Alcazar JJ, Medeiros CB (2008) AI planning in web services composition: a review of current approaches and a new solution. In: *Proceedings of the XXVII Brazilian computer society conference (CSBC)*
56. Juárez-Ramírez R (2017) User-centered design and adaptive systems—toward improving usability and accessibility. *Univ Access Inf Soc* 16(2):361–363
57. Kavcic A (2004) Fuzzy user modeling for adaptation in educational hypermedia. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 34(4):439–449
58. Hwang CL, Yoon K (1981) *Multiple attribute decision making: methods and applications*. Springer, New York
59. Biswas P, Pramanik S, Giri BC (2015) TOPSIS method for multi-attribute group decision-making under single-valued neutrosophic environment. *Neural Comput Appl* 27(3):727–737
60. Open Archives Initiative (2015) The open archives initiative protocol for metadata harvesting. Protocol version 2.0 of 14 June 2002. Document version 8 Jan 2015. <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>
61. Perttunen M, Riekkilä J, Lassila O (2009) Context representation and reasoning in pervasive computing: a review. *Int J Multimedia Ubiquitous Eng* 4(4):1–27
62. Knappmeyer M, Kiani SL, Fra C, Moltchanov B, Baker N (2010) ContextML: a light-weight context representation and context management schema. In: *5th IEEE international symposium on wireless pervasive computing (ISWPC)*, pp 367–372