Preprint submitted to the International Journal of Cloud Applications and Computing, IGI
Find the published version at https://www.igi-global.com/article/towards-a-better-understanding-of-ubiquitous-cloud-computing/196189 , 2018.

Towards a Better Understanding of Ubiquitous Cloud Computing

Hamed Vahdat-Nejad*, Shaghayegh Ostadi Eilaki, Shaghayegh Izadpanah

Pervasive & cloud computing Lab, Faculty of Electrical and Computer Engineering, University of Birjand, Iran

vahdatnejad@birjand.ac.ir, ostadi.sh@birjand.ac.ir, sh.izadpanah@birjand.ac.ir

ABSTRACT

Ubiquitous Cloud Computing has been recently proposed as a new hybrid computing model. This paradigm has two aspects: (a) Making use of cloud services to realize context-awareness (especially for resource-limited mobile devices) and (b) Introducing a new and hybrid computing model. Likewise, this paper mainly involves two subjects. The first is to deeply investigate previous cloud services that have been utilized to design context-aware systems, which highlights the role of cloud computing for realizing context-awareness. The second is to propose an open architecture for the "Ubiquitous Cloud Computing" paradigm. The aim is to technically discuss mutual trends of ubiquitous and cloud computing, to foster the dissemination of state-of-the-art research in this area by providing a reference model for this new computing model, and to present future research directions.

Keywords: Cloud computing, Ubiquitous computing, Context-aware systems

INTRODUCTION

Ubiquitous computing, which is considered as the third era of computing(Abowd, 2016; Krumm, 2016; Weiser & Brown, 1997), seeks ubiquitous provisioning of services to users(Satyanarayanan, 2001). Context is an essential concept in ubiquitous computing, which is defined as "any information that can be used to characterize the situation of an entity"(Dey, Abowd, & Salber, 2001). Context-aware applications, which are usually resided on smart mobile phones, are considered as the building blocks of the ubiquitous computing paradigm(Vahdat–Nejad, Zamanifar, & Nematbakhsh, 2013). These kinds of applications make use of relevant contextual information to provide personalized services to users. Generally, a context-aware application supports several functionalities including context acquisition, modeling, reasoning, and service management(Vahdat-Nejad, 2014), which result in generating massive programs. In spite of recent improvements in hardware technology of smart phones, they suffer from energy, computational and memory limitations comparing with high speed multiple-core laptops and personal computers. However, a major part of context-aware applications that are resided on mobile phones should provide real-time services according to the current situation of the user and environment(Bertolli, Buono, Mencagli, & Vanneschi, 2009). These remarks introduce a paradox that should be resolved before realizing the perspective of ubiquitous computing. In other words, the limited-resource mobile applications have limitations in gathering the contextual information, processing them, and acting in real-time.

Cloud computing has emerged as a new computing model, in which resources can be provided as general services to users, anywhere and on any device(Q. Zhang, Cheng, & Boutaba, 2010). As a result, ubiquitous and on-demand provisioning of services have become an important characteristic of cloud computing. Companies, universities and research centers use cloud computing to acquire virtual resources and improve scalability of applications(Da Cunha Rodrigues et al., 2016; Saurez, Gupta, Mayer, & Ramachandran, 2017). Cloud has previously helped the users and organizations to utilize various resources by a low operational cost(Poorejbari, Vahdat-Nejad, & Mansoor, 2017); Therefore, it can play the role of a supporter for ubiquitous applications. In fact, it seems that cloud computing technology can be exploited to resolve deficiencies of current approaches to realize ubiquitous computing(Shirvani & Vahdat-Nejad, 2016). However, the major part of ubiquitous computing research has been performed independently and separately from cloud computing.

The main contributions of this paper are twofold:

- Firstly, the paper aims to answer whether cloud computing could help to realize ubiquitous computing. In particular, which functionalities of a typical ubiquitous computing system (i.e. context-aware system) could be accomplished by cloud computing. Previously, a few pieces of research have utilized cloud computing services to develop context-aware systems(Grønli, Ghinea, & Younas, 2014; Xiao, Hui, Savolainen, & Ylä-Jääski, 2011). The first contribution of this paper is to investigate these papers for the desired functionalities of a context-aware system that they accomplish by using cloud services. After discussing these functionalities, the question could be answered. An extensive framework is proposed for this purpose. This framework consists of three main dimensions, as the main tasks of a context-aware system, including context acquisition, context processing, and context API and application development. In context acquisition, related papers are reviewed with respect to the cloud usage in context information collection. In context processing, the use of cloud is investigated in processing and making contextual information more operational. Context aggregation, context modeling and context reasoning are different context processing methods. The third dimension includes three parameters of context dissemination, privacy protection and service management. The position of cloud in each of these parameters is investigated, in detail. This helps in understanding the real advantages of cloud computing in current ubiquitous computing systems.

- Although cloud computing and ubiquitous computing have been proposed and formalized individually and separately, they have some mutual features and targets. For example, both of them pursue ubiquitous provisioning of services to the users in an anywhere/anytime manner. As a result, a new computing model, which is referred to as “Ubiquitous Cloud Computing” (UCC), has been recently proposed (Egami, Matsumoto, & Nakamura, 2011; Lomotey & Deters, 2014; Van der Merwe et al., 2010); However, this concept is still vague and in needs of formalization. Specifically, the position of UCC regarding Mobile Cloud Computing (MCC) should be clarified. The other contribution of the paper is to formalize this computing paradigm and propose an open architecture for it. This architecture could illuminate the path of research in this new computing model.

This paper is organized as follows. Section 2 presents related papers that use cloud for performing one or more functional requirements of a context-aware system. In the third section, projects are reviewed with respect to the use of cloud in context acquisition. Section 4 deals with how cloud is used in context processing and relevant parameters. The fifth section investigates the use of cloud in providing supporting platform for context-aware application development. Finally, section 6 discusses the conclusion remarks and open research areas.

PROJECT OVERVIEW

In this section, we overview available projects related to ubiquitous cloud computing, that in some way utilize cloud computing to realize context-awareness. In table 1, we provide a summary of these systems. Afterwards, in the subsequent sections, they are investigated to find the main benefits of the cloud computing for realizing context-awareness.

CasCap(Xiao et al., 2011) is a context-aware power management framework. It takes the advantage of using cloud for processing, storage and networking. In particular, it leverages the cloud to provide secure, low cost and efficient power management for mobile devices. UCM(Yun, Han, Jung, Yeom, & Lee, 2009) aims at creating a tele-management model for managing remote facilities, intelligently, as if they are located locally. It utilizes Cloud platform for tele-management model and provides PaaS. UCM makes use of SmartUM, which is a ubiquitous middleware for managing ubiquitous city. The SmartUM middleware is context-aware and can intelligently process the context information.

UHC(He, Fan, & Li, 2013) uses a private cloud platform architecture, which includes six layers according to its specific requirements. Treating everything as a service and also abstracting a component as a service make it easy to develop this architecture. Six layers of the architecture are: “Services Interaction”, “Services Presentation”, “Session Cache”, “Cloud Engine”, “Medical Data Mining”, and “Cloud Storage”. Besides UC(Egami et al., 2011) aims to provide a platform for ubiquitous service management. The project utilizes cloud computing paradigm to create an environment; that ubiquitously provides all resources as services. It also offers possibility of service management and adaptation.

ICAC(Grønli et al., 2014; Grønli, Hansen, & Ghinea, 2011) takes the advantage of using cloud to store the personal settings and configurations that a user sets for their mobile device. This information is collected through a webpage or local sensor. The project utilizes Google App Engine and has the ability to communicate with Google services like Google calendar, Google contacts, etc. to collect these contextual data. The user's information is stored in Google cloud through their Google account.

Based on the user context information, OVP(Jiang, Wu, Huang, Yang, & Zheng, 2010), CCMR(Otebolaku & Andrade, 2014) and MRUB(Mo, Chen, Xie, Luo, & Yang, 2014) recommend appropriate multimedia files to users. OVP is an online video player which adjusts video formats to codecs running on a user’s mobile phone so that the desired videos can be played without installing a new codec. MRUB is a multimedia recommender system for mobile users in the cloud environment. It reduces the network overhead and accelerates the recommendation process. AML(Karadimce & Davcev, 2013) provides students with the educational content loaded on the cloud based on their context information.

DaaS(Elgazzar, Hassanein, & Martin, 2014) is a web service discovery framework which recommends an appropriate web service in proportion to the user's request and preferences, device limitations, and specific features and environmental parameters. MoCAsH(Hoang & Chen, 2010) uses mobile cloud computing to provide a platform for assistive healthcare systems. This platform is comprised of four main components including mobile agents and sensors, a context-aware mobile cloud middleware, a collaborative cloud computing platform and a cloud portal. The context-aware middleware receives and processes the context information. In addition, this middleware provides some interfaces for user communication and management, collaborative cloud and mobile agents. Users can use cloud services with the help of the cloud portal. ECGAS(Pandey, Voorsluys, Niu, Khandoker, & Buyya, 2012) is a health monitoring system which analyses ECG data in the cloud to provide continuous care for cardiovascular patients. MPHD(Doukas, Pliakas, Tsanakas, & Maglogiannis, 2012) and IOT&CC2PH(Doukas & Maglogiannis, 2012) send medical data to the cloud to let doctors and nurses access data of patients anywhere all the time.

By processing the images of streets in cloud, TLD(Angin, Bhargava, & Helal, 2010) helps blind and low-vision patients to cross the street. CAVS(Wan, Zhang, Zhao, Yang, & Lloret, 2014) is a context-aware vehicular system having three layers: vehicle computational layer, location computational layer, and cloud computational layer. This paper proposes a dynamic parking system, which recommends the available parking spaces on a path and reserves a parking area.

Transportation(Vahdat-Nejad, Ramazani, Mohammadi, & Mansoor, 2016) is an important domain of ubiquitous computing for realizing smart cities(Ramazani & Vahdat-Nejad, 2014, 2017). Cloud2Bubble(Costa, Pitt, Falcão e Cunha, & Galvão, 2012) presents a framework to design and develop intelligent systems in urban environments. The environmental data and passenger data are collected with the sensors embedded in the transportation infrastructure and personal devices such as mobile phones, respectively. Then they are sent to the cloud. User profile and environment status are stored in the cloud in which data processing steps are carried out. The cloud application provides the user with relevant and customized services with respect to user's demands and preferences. For instance, an individual may want to use a train for daily journeys. This framework considers his information and the data of environmental sensors to ascertain whether the temperature is appropriate for him in wagons.

Table 1: Projects Overview

Project	**Organization**	**Publication**		**Situation**
		Journal	Conference	
UCM(Yun et al., 2009)	University of Seoul, South Korea		√	Semi-Implemented
UHC(He et al., 2013)	University of Chinese Academy of Sciences, China	√		Implemented
UC(Egami et al., 2011)	Kobe University Japan		√	Designed
OVP(Jiang et al., 2010)	Tsinghua University, China	√		Designed
CCMR(Otebolaku & Andrade, 2014)	University of Porto, Portugal		√	Implemented
MRUB(Mo et al., 2014)	Huazhong University, China	√		Implemented
AML(Karadimce & Davcev, 2013)	University Ss Cyril and Methodius Skopje, Macedonia		√	Simulated
DaaS(Elgazzar et al., 2014)	Queen's University, Canada	√		Semi-Implemented
AppaaS(Elgazzar, Ejaz, & Hassanein, 2013)	Queen's University, Canada	√		Semi-Implemented
Cloud2Bubble(Costa et al., 2012)	Imperial College London, United Kingdom		√	Implemented
ICAC(Grønli et al., 2014; Grønli et al., 2011)	Norwegian School of IT, Norway	√		Semi-Implemented
CasCap(Xiao et al., 2011)	Aalto University, Finland		√	Semi-Implemented
MoCAsH(Hoang & Chen, 2010)	University of technology, Australia		√	Semi-Implemented
ECGAS(Pandey et al., 2012)	University of Melbourne, Australia	√		Semi-Implemented
TLD(Angin et al., 2010)	Purdue University, USA		√	Semi-Implemented
IOT&CC2PH(Doukas & Maglogiannis, 2012)	University of the Aegean, Greece		√	Semi-Implemented
MPHD(Doukas et al., 2012)	University of the Aegean, Greece		√	Semi-Implemented

CAVS(Wan et al., 2014)	China University of Technology, China	√		Designed

AppaaS(Elgazzar et al., 2013) is a context-aware platform, which provides mobile applications as services. AppaaS uses the user context information, including a user profile or a device profile to provide an appropriate service. The provided platform in this system is meant to match the demands of users with those of providers, and let providers customize their offerings. Users can employ this platform to easily find relevant applications to their situation.

CONTEXT ACQUISITION

A context-aware system usually performs actions when some contextual variables meet specific values. Therefore, it should be able to discover required context sources. A context source is defined to be an entity that produces raw contextual data(Vahdat-Nejad, 2014). Usually, context-aware systems provide registration mechanisms for context sources to declare their contextual capacities. Some applications need a history of particular contexts, for various purposes. This motivates the need for storage of contextual data. This storage is also necessary for context mining and pattern discovery purposes. Because most sensors and devices have a limited storage memory, and considering the vast and increasing volume of produced context in a large-scale environment, Context acquisition becomes a challenging issue. Some mechanisms have already been proposed(Fonteles, Neto, Maia, Viana, & Andrade, 2013; Kramer, Kocurova, Oussena, Clark, & Komisarczuk, 2011) for context acquisition in order to reduce the complications of developing context-aware applications. In the majority of previous cloud-based context-aware systems, the process of context acquisition is conducted by the cloud.

In this case, the use of cloud can help to meet storing challenges. In MRUB and CCMR, the media descriptor information is collected by the cloud. This context information includes media genre, video length, resolution, and video category. Part of these contextual elements are available on the IMDB database("Internet Movie Database(IMDb),"). The cloud uses this database to collect the contextual information about the media. In addition to the media contextual information, the social content information is loaded on the cloud in MRUB. This information includes the user favorite communities, the user idol list (including the artists and individuals whose posts have many fans), and the user favorite videos.

In the healthcare applications(Rahnama-ye-Moqaddam & Vahdat-Nejad, 2015), a great portion of contextual information can be problematic. Considering the cloud storage, the data collected from sensors can be stored in the cloud (such data usually includes vital signs)(Poorejbari & Vahdat-Nejad, 2014). Moreover, relevant users such as doctors, nurses and caregivers can access these data elements anywhere all the time; therefore, patients receive a continuous care. In ECGAS, the numerical data and ECG graphs are stored in the cloud. The contextual information includes the user-id, age, gender and sleeping time. In this application, ECG graphs are sampled. Then this information is used to draw and store ECG waveform and graph in the cloud. Doctors and nurses can use such data to check the status of cardiovascular patients. These data are always accessible via cloud. Similarly, In UHC, medical data needs to be uploaded to the cloud. First, physiological data collected by sensors are transmitted to a gateway using USB or Bluetooth technology, and then the data from the gateway is sent to the cloud server. Moreover, the top layer of architecture, namely, “Service Interaction” allows users to upload the physiological data through mobile phones, computers, etc. to the cloud server.

User profiles include a great portion of contextual information that can describe user needs. In AML, users state in their profile that whether they like to receive the educational content visually or in audio form. Users' profiles are stored in the cloud. They are used for next decisions and appropriate services. In Cloud2Bubble, users specify the appropriate values of some context elements in their profiles. For instance, a user may set 27 degrees as an appropriate environment temperature. Therefore, the system provides the user an appropriate response by comparing the current context and the user profile stored in the cloud. In ICAC, the contextual information can be collected in two different ways. The first is through cloud services that are implemented as webpages, which are owned by users. Users enter personal contextual information such as calendar and contact list on the webpages that transmit them to the cloud server. When the user logs in to the system for the first time, all his/her personal configurations are stored in the cloud. Users can also change their personal configurations via webpages. The other method is to collect contextual information by using local sensors of mobile devices, and then transmit it to the cloud for storage and retrieval.

UC makes use of the “context manager” component to gather contextual data from sensors and context sources. This component is designed in the platform layer of cloud computing (PaaS). Other components of the system extensively utilize the contextual information acquired by this component. Besides, CasCap exploits the “network mapping service” as a cloud SaaS to collect the contextual information of mobile devices. To this end, the “context manager” which is a local component on mobile devices, collects information from built-in sensors of the mobile device. Afterward, the context information is uploaded to the cloud.

Sometimes, the process of collecting and storing context is more complicated. In DaaS, the cloud is used to collect the contextual information on the descriptors of services. Service providers create documents to describe services. Then these documents are stored in a directory. Using a service engine, the cloud crawls into directories. The document describing an appropriate service is stored in the cloud after identification. These documents are classified with the service clustering components and then stored in a repository.

Finally, In UCM, the SmartUM middleware is deployed in the platform layer of cloud and is responsible for receiving data from different local sensors that are located in the feeling tier of the UCM architecture. SmartUM manages all remote sensors, which are located in a network. This network is called Ubiquitous Sensor Network (USN). Contextual data received from USN is sent to Context-aware Computing Layer (CCL) for later actions. The contextual information is then stored in the context repository, which gives the permission to other components to query, modify, add, or delete context elements. According to table 2 that summarizes these systems, most of them use cloud for context acquisition and storage. Therefore, it seems that cloud can play an important role in context acquisition. Although, at first cloud may seem a desirable solution for collecting rather static contextual information such as media descriptors (e.g. genre, length, resolution, and category), it also helps in gathering dynamic contextual information such as physiological data obtained from medical wearable sensors (e.g. vital signs). As a result of ubiquity of wireless networks as well as high communication speed and low delays, in the future cloud could play a dominant role for context acquisition.

CONTEXT PROCESSING

The data obtained from sensors are raw and low-level. They should be processed to be used better. Processing is sometimes done before storing, something which decreases the size of stored data. In some cases, some techniques like ontology is used to model the context data. There are different methods for context processing, which are described in the following. Then the projects that use the cloud to do these processes are reviewed.

• CONTEXT AGGRIGATION

One of the common processing methods is context aggregation. When the amount of raw data obtained from sensors is large, it might be useless to store all of them. Context aggregation can create significant meaningful context information(Raychoudhury, Cao, Kumar, & Zhang, 2013). This information can be easily transferred and stored due to low size. For instance, a great amount of raw data are produced in a wireless sensor network. It takes too much memory to store all of them over time. A context aggregation system can use the parameters of quality of context to eliminate duplicates and conflicts(Manzoor, Truong, & Dustdar, 2009). Context aggregation may impose too much overhead on the system and drain the battery(Gu, Pung, & Zhang, 2005; Ranganathan & Campbell, 2003). Some methods have been proposed to solve this problem including the distributed context aggregation. For instance, EDCI(Tan, Zhang, Wang, & Cheng, 2005) distributes the aggregation process to reduce the time and Solar(Chen & Kotz, 2002) is focused on increasing the reusability to avoid duplicates. Besides, the cloud is highly capable of storing and computing, a fact which can be a solution to the abovementioned problem. In spite of that, only a few systems make use of cloud for context aggregation, as follows.

AppaaS uses a scoring service to select applications. The total scores of users are regarded as the context information in the selection process. Each user scores an application that he/she uses. The total score of each application is calculated among similar applications. Then an application with the highest score is selected. Collecting and averaging these scores are done in the cloud. Like Appaas, DaaS uses the scoring service to select an appropriate web service. User scores indicate user satisfaction and the quality of service. Although keeping all scores of users may be primarily useless; aggregated scores can provide users and developers with more useful information.

• CONTEXT MODELING

Raw contextual data produced by sources (e.g. sensors) should be modeled and transformed to meaningful information to be usable by applications. Modeling is concerned with two aspects, which are explained by examples: (a) Assume an application requires information about the noise level of a specific room. A noise measuring sensor provides a digital value for this context, but the programmer needs to know whether the noise level is silent, low, medium or high. (b) Another noise measuring sensor exists for another room, with the sensor constructed by another company that follows another standard for representation of results. The application needs to find the room with the lower noise level, but due to different standards, numerical results of the sensors could not be directly compared. Therefore both of them should employ a standard understandable by the programmer. Previously**,** some context models have been proposed (Baldauf, Dustdar, & Rosenberg, 2007; Strang & Linnhoff-Popien, 2004; Truong & Dustdar, 2009), but the main approaches to context modeling are object-role based, spatial models and ontology- based methods(Bettini et al., 2010). The main issue in context modeling is to reach a unique standard for facilitating interactions among different context-aware systems. It seems that cloud could play a leading role for this purpose.

Several systems make use of cloud for context modeling. In MoCASH, the data measured with different sensors are sent to the context-aware middleware, which is embedded in the cloud. One of the components of this middleware is a context convertor. The context convertor converts the collected raw context data to the usable context information. Then high-level contextual information is provided to a context analyzer. In this architecture, each context element is defined as an array of [ID, Values]. Through this ID, the context element is processed by different subcomponents of the context analyzer.

In ICAC, each contextual information is modelled and described by a function in the cloud. For instance, the user's social settings are shown as a set of meta-tags. They are defined as $[type = 'tag name']. For example, if a tag is defined as $[type = work], it means that the user is working, and only work-related calls could be connected. These tags can be changed by the web interfaces of Google contacts and calendar.

As pointed out earlier, an ontology is one of the most important methods for context modelling. In OVP, the information collected from clients is modelled with an ontology in the cloud. The purpose is to display the complicated context data well in order to facilitate reasoning and sharing. An ontology is able to represent a set of concepts and their relationships within a domain. It also facilitates the semantic interoperability.

In MRUB, the features of a video are modelled in a list or vector of attributes inside the cloud. Each attribute indicates one feature of the video. User connections are modelled with a graph. This graph is comprised of three subgraphs including idol fan, co-commenting behaviors, and interest group. The idol-fan subgraph shows the relationship between the user and other individuals whom he/she has chosen as his/her idol. If a user makes a comment on a favorite video, an edge will be created between the user and video in the co-commenting behaviors subgraph, which can be exploited for obtaining the similarity between users. The interest groups subgraph displays the groups in which the user is a member. The user-user community graph results from the combination of these three subgraphs. It shows a community of users who like similar videos.

Finally, in CasCap, the crowd-sourced context monitoring services run in the cloud. They have the responsibility of modeling context information of mobile phones and other devices available in the network. These services process the gathered context information into more useful and expressive form. In summary, these systems practically prove that cloud could be regarded as an important base for performing context modeling and providing interaction between various applications and users.

- **CONTEXT REASONING**

Sensors can only measure simple context types. It is not possible to directly measure a high-level context such as activity of a person or a complicated event such as organizing a meeting. In addition, deriving implicit contextual information needs sophisticated techniques and time-consuming algorithms. In these situations, a reasoning component is exploited to derive high-level contexts from low-level ones. Some context reasoning techniques are logic-based(Ranganathan & Campbell, 2003), case-based(D. Zhang, Cao, Zhou, & Guo, 2009), Bayesian network-based(Mamei & Nagpal, 2007; Ranganathan, Al-Muhtadi, & Campbell, 2004)and ontology-based(Ejigu, Scuturici, & Brunie, 2007; Nieto, Gutiérrez, & Lancho, 2009). Because a context reasoning component usually requires different local and global pieces of information, and executes time-consuming algorithms, it is regarded as a challenging issue to be performed on mobile devices of a context-aware system. Cloud as a rich resource of computation can be a serious candidate for performing context reasoning. Among the systems, a significant number make use of cloud for context reasoning.

In Cloud2Bubble, context reasoning on user profile, environmental conditions, user preferences, emotional inputs and user's quality of experience is required to offer customized services(Beauregard, Younkin, Corriveau, Doherty, & Salskov, 2007). All the steps of reasoning and presenting an appropriate solution are done in the cloud. The decision making process of CAVS depends on different factors such as road condition, weather condition and traffic flow forecasting. By collecting data from vehicular social network and mining them, high-level contextual information such real-time and historical traffic flow capacity can be obtained. This kind of information helps in effective prediction and decision making processes. This context information is used for dynamic parking service.

TLD uses image processing algorithms and AdaBoost(Freund & Schapire, 1997) to detect the status of traffic lights. AdaBoost is an adaptive machine learning algorithm, which can detect objects very fast. In this system, the traffic light detector uses a classification method based on AdaBoost to detect the status of the traffic light with the help of video frames. The image processing operations are time-consuming; hence, video frames are processed in the cloud. In the proposed framework of DaaS, web services are specified according to a set of keywords extracted from descriptive files. The user's request is sent to the cloud in a simple text file. In the cloud, a component named service request handler does the extraction and inference of keywords and meaningful information from the user's request with the help of WSDL. Then the search/matchmaking component matches these keywords with the descriptors of web services in WSDL to find the appropriate web service.

User preferences are required for next recommendations in CCMR and are deducted in the cloud. To this end, a tree is created in each user's profile. The root of this tree includes the user ID. The second-level and the third-level nodes include the media category and genre, respectively. Therefore, more accurate context information are obtained on the media. This information can show whether the film is modern or classic and what its language is. The leaves of tree includes user context information. Each time the user runs the application and watches media in a specific category or genre, the weight of the connected edge to this corresponding leave is increased in the tree. This weight indicates user preferences. It is used in the next recommendation process. All the steps of simulating this tree are run in the cloud. The similarity of users is also taken into consideration in media recommendation. The processes of evaluating the similarity of users are also run in the cloud.

Reasoning and analysis are done by a component named context analyzer in MoCAsH. This component and other dependent components are established in the cloud. The context analyzer classifies the defined contextual information and delivers them to the appropriate module. Concurrently, context rules are loaded from the context repository. The context repository keeps the files of dynamic rules, context information and context sources. Based on these rules, the context analyzer considers the new context data and presents a relevant service. The reasoning system has several subcomponents for analysis in the context analyzer. They are healthcare, network quality of service and application power management.

In UCM, The CCL (which lies in platform layer of cloud) is responsible for processing the received contextual information to provide high-level intelligent context by using specific domain ontologies. The reasoning is performed by using inference engine available in context analyzer component. For supporting various kinds of reasoning tasks, multiple inference engines can be established and installed. Generally UCM uses rule-based approach based on first-order logic for context reasoning. Table 2 also reviews the projects with respect to context processing. As can be seen cloud platform has been widely used for context reasoning. The reason is twofold. Context reasoning is a common operation in today's context-aware systems. Without reasoning, context-aware systems could not show enough intelligence to attract users; however, it is considered as a resource-intensive process, hence cloud is the major candidate for performing it. On the other hand, cloud computing is hardly used for context aggregation, because context aggregation is not a common process in context-aware systems and only a few projects exploit it. Finally, there are several approaches to context modeling and part of them are light-weight modules, which could be performed on mobile devices.

Table 2: The summary of cloud-based context acquisition and processing

Project	**Context Acquisition**	**Context Processing**		
		Context Aggregation	Context Modeling	Context Reasoning
UCM(Yun et al., 2009)	√	×	×	√
UHC(He et al., 2013)	√	×	×	×
UC(Egami et al., 2011)	√	×	×	×
OVP(Jiang et al., 2010)	×	×	√	×
CCMR(Otebolaku & Andrade, 2014)	√	×	×	√
MRUB(Mo et al., 2014)	√	×	√	×
AML(Karadimce & Davcev, 2013)	√	×	×	×
DaaS(Elgazzar et al., 2014)	√	√	×	√
AppaaS(Elgazzar et al., 2013)	×	√	×	×
Cloud2Bubble(Costa et al., 2012)	√	×	×	√
ICAC(Grønli et al., 2014; Grønli et al., 2011)	√	×	√	×
CasCap(Xiao et al., 2011)	√	×	√	×
MoCAsH(Hoang & Chen, 2010)	×	×	√	√
ECGAS(Pandey et al., 2012)	√	×	×	√
TLD(Angin et al., 2010)	×	×	×	√
IOT&CC2PH(Doukas & Maglogiannis, 2012)	×	×	×	×
MPHD(Doukas et al., 2012)	×	×	×	×
CAVS(Wan et al., 2014)	×	×	×	√

CONTEXT API & APPLICATIONS DEVELOPMENT

Nowadays context-aware applications can show the extensive abilities of pervasive computing and provide users with flexible and efficient applications. Context-aware applications can perform user activities on behalf of them. Despite all of these unique features, context-aware applications have not been able to completely enter the market yet. High deployment overhead and security concerns, especially privacy, are among the significant and challenging issues in this regard(Henricksen & Indulska, 2004). Cloud computing can help to meet these challenges. The use of cloud for context dissemination, privacy protection and service management can facilitate the implementation and deployment of context-aware applications. In continue, we investigate and review the usage of cloud for accomplishing these tasks.

- **CONTEXT DISSEMINATION**

Context dissemination includes the methods of delivering context to consumers. From the consumer's perspective, this can be named context acquisition, too(Perera, Zaslavsky, Christen, & Georgakopoulos, 2014). However, context dissemination can be regarded as a middleware functionality, which makes it possible to inject contextual information to the system and delivers it to all the interested entities, automatically(Bellavista, Corradi, Fanelli, & Foschini, 2012). Two common methods of context dissemination are query-based and event-based. In the query-based method, a consumer issues a request as a query to the system. In the event-based method, a consumer is subscribed to a context source or an event. Then the system publishes the context periodically or when a new context value creates. This method is named publish/subscription.

In the two instances of investigated systems, the cloud is responsible for context dissemination. In MPHD, physiological information related to healthcare programs is sent to the cumulocity cloud. Afterward, the contextual information is available to web-based applications and mobile applications. The cloud provides users and applications with APIs to be able to access the desired context elements.

In the IOT&CC2PH system, context dissemination is performed through communication APIs offered by the cloud. In this system, lightweight APIs are used so that sensors can easily send data. Using these APIs, external applications can access the contextual information of sensors through the cloud. External applications can also use these APIs to process context and manage the alarm system.

- **PRIVACY PROTECTION**

Security has been a fundamental concern in both cloud(Bhushan & Gupta, 2017; S. Gupta & Gupta, 2016) and ubiquitous(B. B. Gupta, Gupta, & Chaudhary, 2017; S. Gupta & Gupta, 2017) computing paradigms. Sometimes, context information includes individuals' sensitive information, which can result in security threats. Therefore, developers should employ a mechanism to protect personal information and privacy of users. Some privacy protection techniques include access control, anonymity and cryptography(B. Gupta, Agrawal, & Yamaguchi, 2016). Running the processes of these techniques may sometimes need too much time and high performance servers. Therefore, cloud capabilities can be used for this purpose. This subsection deals with the systems that use the cloud to protect privacy.

In MPHD, an anonymization approach is used to protect privacy, and data elements are stored without names in the cloud. Moreover, sensors and the cloud are connected under SSL protocol in which each sensor has a confidential key by which it is authenticated. The process of logging into the system in ICAC is done with a Google username and password. Open authorization can also make it possible to use some tokens as authentication tools so that the application can be accepted as the third party which is authorized to access the information. Therefore, the sharing of private information of calendar and contacts is facilitated on the Google cloud.

In OVP, a component named corsair has been developed on the cloud. It keeps and manages the metadata of videos such as content description, context information received from users as well as the authentication information including usernames and passwords. In corsair, the memory storage is divided into three parts: private space, public space and community space. In the private space, only the user has access to the information. However, all the individuals access data in the public space. In the community space, data is visible only to the community members. Therefore, a community can be created to authorize only the members to access the desired information. In this system, the privacy protection approach is access control, and the user monitors the way in which their private information is used. In cascap, dedicated clones are used for each user. These clones are located in the cloud. One clone is dedicated to each mobile device, and the context information of users are sent only to their own dedicated clones. The designers of this system believe that this method operates better than the proxy-based method because one clone is dedicated to only one device, but a proxy is shared among a number of devices.

- **SERVICE MANAGMENT**

In a smart environment diverse types of basic services exist(Vahdat-Nejad, Zamanifar, & Nematbakhsh, 2013). A service can be a software/hardware service for controlling a device (such as starting fire alarm, showing daily news on the screen), or a basic context-aware service (such as turning off lights in a building after everyone has left for the day), or even a complex context-aware application. Application developers usually search for these miniature service components and try to build their context-aware applications by composing suitable available services(Vahdat-Nejad, 2014) Ubiquitous computing paradigm should provide the functionality for service discovery and composition. Directory-based and DHT-based approaches are popular for traditional service discovery. Besides, there should be a way for service representation by service providers and service request by users(Vahdat-Nejad, 2014). On the other hand, most of the devices available in the ubiquitous computing environment are mobile phones and PDAs with limited memory, computational, communication and availability capacities(Bertolli, Buono, et al., 2010; Bertolli, Mencagli, & Vanneschi, 2010). They cannot play the role of high performance computers and also may sometimes be inaccessible or even off. Therefore, service management is regarded as a challenge in this environment and there are only a few preliminary works in this area(Vahdat-Nejad, 2014).

Previously, a few systems use the cloud for service management. AppaaS tries to recommend the most appropriate mobile application to the user in accordance with user context information. Context information includes location, user profile, device profile, user score and time. AppaaS enables App developers to improve the functional control of their applications dynamically with the help of the cloud. AppaaS does the recommendation of mobile applications as a cloud service. It also manages these services by applying access restrictions.

The cloud service interface is a component responsible for service management in MoCAsH. It operates as the entrance of cloud services. The physiological context information is analyzed by a context broker in the cloud in order to find emergency situations. In such situations, relevant services are invoked, automatically.

In UC, the cloud service providers register their services to the "Service resource registry" component, which is defined as an IaaS. This component stores metadata explaining services. Application developers make use of this component to find required services. For this, they can use a variety of attributes including device class, operation type, physical location, purpose, users, etc. For simplicity, another component which is called "Adaptive resource finder" is proposed in the platform layer of cloud computing to recommend appropriate services to users based on their requirements and contexts. It understands and transforms requirements to queries for the "Service resource registry", and makes use of the context manager component to filter results according to the contextual information of users. Table 3 classifies the investigated projects from the perspective of context API and App development. Although at first, it may seem that cloud could be a suitable solution for overcoming the challenges of context dissemination, privacy protection, and service management, only a minority of these systems make use of it. The main reason is that these parameters are the newest among the functionalities of a context-aware system; therefore, there is much less research on them in comparison with context acquisition and processing. In fact, complete realization of context dissemination, privacy protection, and service management leads to a real pervasive computing environment. Today, we are far from that ideal point.

Table 3: Cloud-based context API and application development parameters

Project	Context Dissemination	Privacy Protection	Service Management
UCM(Yun et al., 2009)	×	×	×
UHC(He et al., 2013)	×	×	×
UC(Egami et al., 2011)	×	×	√
OVP(Jiang et al., 2010)	×	√	×
CCMR(Otebolaku & Andrade, 2014)	×	×	×
MRUB(Mo et al., 2014)	×	×	×
AML(Karadimce & Davcev, 2013)	×	×	×
DaaS(Elgazzar et al., 2014)	×	×	×
AppaaS(Elgazzar et al., 2013)	×	×	√
Cloud2Bubble(Costa et al., 2012)	×	×	×
ICAC(Grønli et al., 2014; Grønli et al., 2011)	×	√	×
CasCap(Xiao et al., 2011)	×	√	×
MoCAsH(Hoang & Chen, 2010)	×	×	√
ECGAS(Pandey et al., 2012)	×	×	×
TLD(Angin et al., 2010)	×	×	×
IOT&CC2PH(Doukas & Maglogiannis, 2012)	√	×	×
MPHD(Doukas et al., 2012)	√	√	×
CAVS(Wan et al., 2014)	×	×	×

ARCHITECTURAL DESIGN

According to tables 2 and 3, cloud computing has been recently utilized to realize various functionalities of a context-aware system. Context-aware systems are regarded as the building blocks of a ubiquitous computing environment. Besides, cloud and ubiquitous computing have some similarities. For instance, both of them aim to provide services pervasively to the users. As a result, the concept of Ubiquitous Cloud Computing (UCC) has been recently introduced (Lomotey & Deters, 2014).

The proposed architecture of UCC lies in three layers as shown in figure 1. In continue, components of each layer are described.

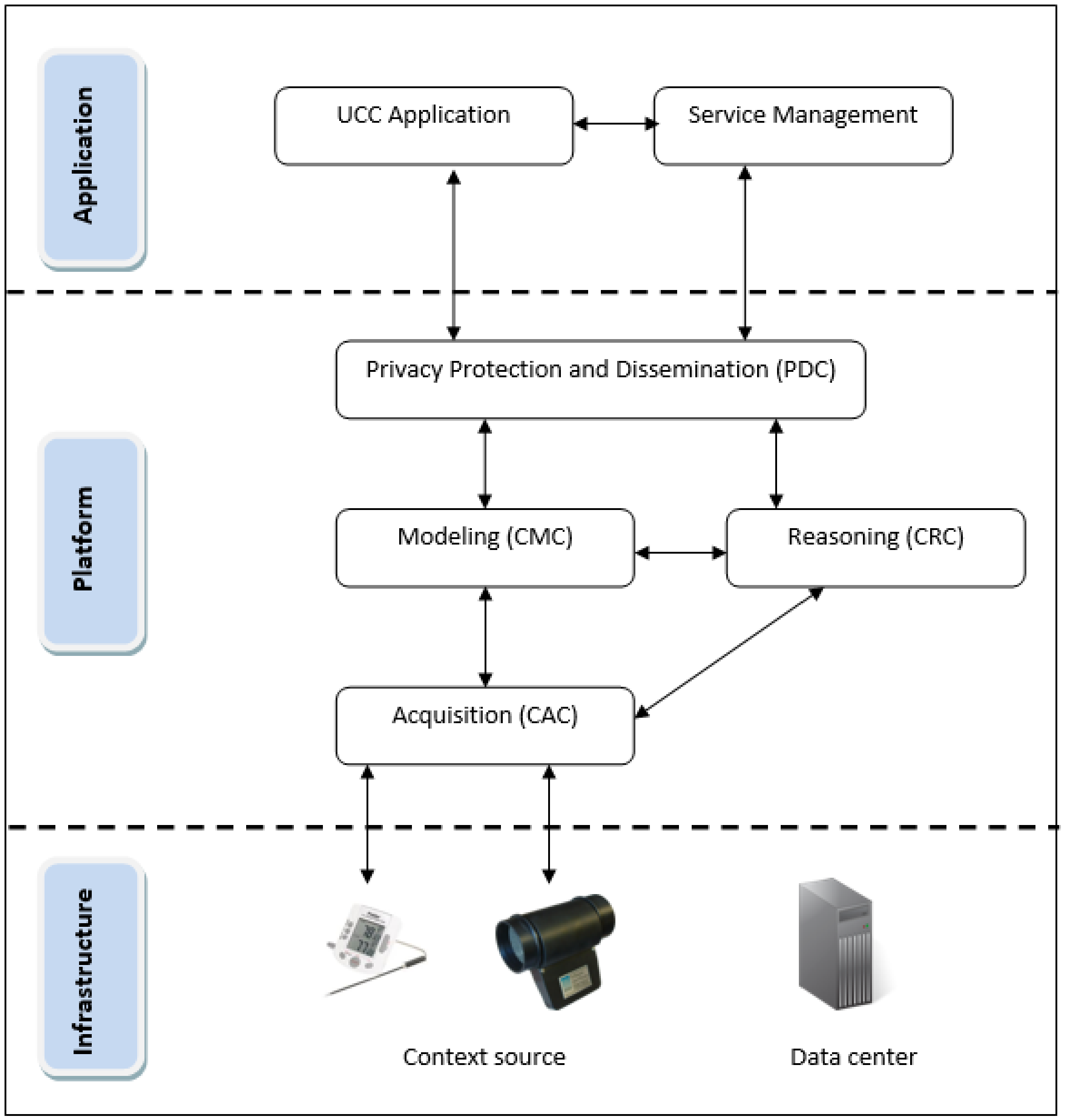

Figure 1. Ubiquitous cloud computing architecture

- *The infrastructure layer*: This layer consists of hardware infrastructure of UCC. Data centers and context sources are main hardware of this layer. In general, context sources involve sensors and embedded computers that produce contextual data for being explored by applications.
- *The platform layer*: The platform layer is built on top of the infrastructure layer. In addition to the functional tasks of this layer in Mobile Cloud Computing (MCC) such as providing operating systems, middleware APIs and programming frameworks, the platform layer aims to gather contextual data from diverse

sources and transform it to meaningful contextual information. The purpose of this layer is to minimize the burden of acquiring required context elements for UCC applications. It especially gives benefit to the real-time applications and also to the applications residing on limited resource devices such as mobile phones. The platform layer consists of four components:

✓ Context Acquisition Component (CAC): CAC is responsible for discovering context sources and for gathering generated raw contextual data.

✓ Context Modeling Component (CMC): This component transforms raw contextual data (e.g. meaningless digits) to meaningful contextual information (e.g. variables) to be useful by applications. Providing context modeling as a PaaS in the cloud, helps reaching better understanding and interactions among different context-aware systems.

✓ Context Reasoning Component (CRC): Context reasoning is usually performed by exploiting time–consuming algorithms such as Bayesian, Fuzzy, and Ontology reasoning to deduce high-level context like global traffic congestion level of streets of a city. Many real-time applications require these information values. For example, collision avoidance or navigation applications in the vehicular network domain typically work according to street's traffic congestion level. CRC is responsible for reasoning high-level contextual information by using low-level context types.

✓ Privacy protection and context Dissemination Component (PDC): PDC is responsible for distributing contextual information among interested UCC users and applications. Because many context elements are private information, PDC follows defined policies of owners for context dissemination. Finally, it provides Application Programming Interfaces (API) for programmers to be able to obtain pure contextual information.

- *The application layer*: At the highest level, the application layer is mainly consists of two parts: (a) service management component, and (b) actual UCC applications. Service management component is responsible for service discovery and composition. Application developers can utilize service management and also PDC to reduce efforts for developing their UCC applications.

CONCLUSION and FUTURE RESEARCH

In this paper, we have investigated the new paradigm of ubiquitous cloud computing. At first, available related research has been reviewed according to the mentioned functionalities of a context-aware system that are realized by cloud computing services. In general, cloud computing has been used for realizing context acquisition, processing and dissemination as well as application development. However, since context dissemination and service management are rather newer issues in pervasive computing, less research has been performed on it. On the other hand, context acquisition is a common functionality of pervasive systems and many cloud-based systems have been proposed to accomplish it. Among context processing techniques, reasoning is the most resource-intensive and common functionality, hence the most cloud-based systems have concentrated on it. Afterward, the architecture of the new paradigm of ubiquitous cloud computing has been proposed and discussed. Similar to the prevalent layers of cloud computing architecture, it has three layers of infrastructure, platform, and application, which aggregate ubiquitous and cloud computing paradigms. In summary, ubiquitous cloud computing is in early research stages and needs many research works.

Challenges of realizing ubiquitous cloud computing are categorized into two levels. The high-level considers architectural issues associated to designing a UCC system. In this stage at the higher level, all requirements of this computing model should be considered and satisfied explicitly or implicitly through components and their interactions; hence it highly affects the capabilities of the UCC model. The main requirements and limitations that should be considered in this stage include need for distribution, resource limitations of devices, dynamic nature of the environment, and mobility of entities.

The lower level concerns single functionalities of UCC including context acquisition, modeling, reasoning, dissemination, service management and privacy protection. Although, many works have been performed on local context acquisition, modeling and reasoning in ubiquitous computing domain, there are only a few preliminary studies about them in the UCC domain. Service management is rather a new research subject in ubiquitous computing that needs much more attention in UCC.

The goal of the UCC is to support the development of ubiquitous cloud applications. One of the benefits and services of the UCC to the applications is provided and seen through context dissemination mechanisms. Providing extensible software tools including easy-to-use and transparent APIs from the view of application developers is important for realizing UCC. On the other hand, privacy is an important issue to achieving the ubiquitous cloud computing vision. However, most of the studies thoroughly ignore or just support a simple access control mechanism. Privacy protection while disseminating contextual information is a challenging research subject in UCC environments**.**

The proposed architecture is suitable for computing-intensive as well as data-intensive UCC applications; however, real-time applications may suffer from the latency of the WAN communications. To address this issue, for the next stage of this research the impacts of the edge and fog computing paradigms for resolving this drawback will be investigated. In particular, a new architecture should be proposed to integrate the main components of edge and fog computing into the proposed three-layered UCC architecture.

Acknowledgement

It is a pleasure for authors to thank Mr. Hussein Shirvani and Ms. Narjes Hajizadeh for their valuable remarks.

REFERENCES

Abowd, G. D. (2016). Beyond weiser: From ubiquitous to collective computing. *Computer, 49*(1), 17-23.

Angin, P., Bhargava, B., & Helal, S. (2010). *A mobile-cloud collaborative traffic lights detector for blind navigation.* Paper presented at the International Conference on Mobile Data Management Kansas City, MO, USA.

Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing, 2*(4), 263-277.

Beauregard, R., Younkin, A., Corriveau, P., Doherty, R., & Salskov, E. (2007). Assessing the Quality of User Experience. *Intel Technology Journal, 11*(1).

Bellavista, P., Corradi, A., Fanelli, M., & Foschini, L. (2012). A survey of context data distribution for mobile ubiquitous systems. *ACM Computing Surveys (CSUR), 44*(4), 24.

Bertolli, C., Buono, D., Mencagli, G., Torquati, M., Vanneschi, M., Mordacchini, M., & Nardini, F. M. (2010). *Resource discovery support for time-critical adaptive applications.* Paper presented at the Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, Caen, France.

Bertolli, C., Buono, D., Mencagli, G., & Vanneschi, M. (2009). *Expressing Adaptivity and Context Awareness in the ASSISTANT Programming Model.* Paper presented at the Autonomics, Limassol, Cyprus.

Bertolli, C., Mencagli, G., & Vanneschi, M. (2010). *Analyzing memory requirements for pervasive grid applications.* Paper presented at the 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Pisa, Italy.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and mobile computing, 6*(2), 161-180.

Bhushan, K., & Gupta, B. (2017). Security challenges in cloud computing: state-of-art. *International Journal of Big Data Intelligence, 4*(2), 81-107.

Chen, G., & Kotz, D. (2002). *Context aggregation and dissemination in ubiquitous computing systems.* Paper presented at the IEEE Workshop on Mobile Computing Systems and Applications, Callicoon, NY, USA.

Costa, P. M., Pitt, J., Falcão e Cunha, J., & Galvão, T. (2012). *Cloud2Bubble: enhancing quality of experience in mobile cloud computing settings.* Paper presented at the ACM workshop on Mobile cloud computing and services, Low Wood Bay, Lake District, UK.

Da Cunha Rodrigues, G., Calheiros, R. N., Guimaraes, V. T., Santos, G. L. d., de Carvalho, M. B., Granville, L. Z., . . . Buyya, R. (2016). *Monitoring of cloud computing environments: concepts, solutions, trends, and future directions.* Paper presented at the Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy.

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-computer interaction, 16*(2), 97-166.

Doukas, C., & Maglogiannis, I. (2012). *Bringing iot and cloud computing towards pervasive healthcare.* Paper presented at the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing Palermo, Italy.

Doukas, C., Pliakas, T., Tsanakas, P., & Maglogiannis, I. (2012). Distributed Management of Pervasive Healthcare Data through Cloud Computing *Wireless Mobile Communication and Healthcare* (pp. 386-393). Berlin, Heidelberg: Springer.

Egami, K., Matsumoto, S., & Nakamura, M. (2011). *Ubiquitous cloud: Managing service resources for adaptive ubiquitous computing.* Paper presented at the International Conference on Pervasive Computing and Communications Workshops Seattle, WA, USA.

Ejigu, D., Scuturici, M., & Brunie, L. (2007). *An ontology-based approach to context modeling and reasoning in pervasive computing.* Paper presented at the International Conference on Pervasive Computing and Communications Workshops White Plains, NY, USA.

Elgazzar, K., Ejaz, A., & Hassanein, H. S. (2013). AppaaS: offering mobile applications as a cloud service. *Journal of internet services and applications, 4*(1), 1-12.

Elgazzar, K., Hassanein, H. S., & Martin, P. (2014). Daas: Cloud-based mobile web service discovery. *Pervasive and mobile computing, 13*, 67-84.

Fonteles, A. S., Neto, B. J., Maia, M., Viana, W., & Andrade, R. M. (2013). An adaptive context acquisition framework to support mobile spatial and context-aware applications *Web and wireless geographical information systems* (pp. 100-116). Berlin, Heidelberg: Springer.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences, 55*(1), 119-139.

Grønli, T.-M., Ghinea, G., & Younas, M. (2014). Context-aware and automatic configuration of mobile devices in cloud-enabled ubiquitous computing. *Personal and ubiquitous computing, 18*(4), 883-894.

Grønli, T.-M., Hansen, J., & Ghinea, G. (2011). Integrated context-aware and cloud-based adaptive home screens for android phones *Human-Computer Interaction. Interaction Techniques and Environments* (pp. 427-435). Berlin, Heidelberg: Springer.

Gu, T., Pung, H. K., & Zhang, D. Q. (2005). A service-oriented middleware for building context-aware services. *Journal of Network and computer applications, 28*(1), 1-18.

Gupta, B., Agrawal, D. P., & Yamaguchi, S. (2016). *Handbook of research on modern cryptographic solutions for computer and cyber security*. Hershey, Pennsylvania: IGI Global.

Gupta, B. B., Gupta, S., & Chaudhary, P. (2017). Enhancing the Browser-Side Context-Aware Sanitization of Suspicious HTML5 Code for Halting the DOM-Based XSS Vulnerabilities in Cloud. *International Journal of Cloud Applications and Computing (IJCAC), 7*(1), 1-31.

Gupta, S., & Gupta, B. (2016). XSS-secure as a service for the platforms of online social network-based multimedia web applications in cloud. *Multimedia Tools and Applications, 75*(14), 1-33.

Gupta, S., & Gupta, B. (2017). Detection, Avoidance, and Attack Pattern Mechanisms in Modern Web Application Vulnerabilities: Present and Future Challenges. *International Journal of Cloud Applications and Computing (IJCAC), 7*(3), 1-43.

He, C., Fan, X., & Li, Y. (2013). Toward ubiquitous healthcare services with a novel efficient cloud platform. *Biomedical Engineering, IEEE Transactions on, 60*(1), 230-234.

Henricksen, K., & Indulska, J. (2004). *A software engineering framework for context-aware pervasive computing.* Paper presented at the IEEE Annual Conference on Pervasive Computing and Communications, Orlando, FL, USA.

Hoang, D. B., & Chen, L. (2010). *Mobile cloud for assistive healthcare (MoCAsH).* Paper presented at the Services Computing Conference Hangzhou, China.

Internet Movie Database(IMDb). (2017). 2017, from www.imdb.com
Jiang, J., Wu, Y., Huang, X., Yang, G., & Zheng, W. (2010). Online video playing on smartphones: A context-aware approach based on cloud computing. *Journal of Internet Technology, 11*(6), 821-828.
Karadimce, A., & Davcev, D. (2013). *Adaptive Multimedia Learning Delivered in Mobile Cloud Computing Environment.* Paper presented at the International Conference on Cloud Computing, GRIDs, and Virtualization Valencia, Spain.
Kramer, D., Kocurova, A., Oussena, S., Clark, T., & Komisarczuk, P. (2011). *An extensible, self contained, layered approach to context acquisition.* Paper presented at the International Workshop on Middleware for Pervasive Mobile and Embedded Computing, Lisbon, Portugal.
Krumm, J. (2016). *Ubiquitous computing fundamentals*. Redmond, Washington: CRC Press.
Lomotey, R. K., & Deters, R. (2014). *Architectural designs from mobile cloud computing to ubiquitous cloud computing-survey.* Paper presented at the 2014 IEEE World Congress on Services, Anchorage, AK, USA.
Mamei, M., & Nagpal, R. (2007). *Macro programming through Bayesian Networks: Distributed inference and anomaly detection.* Paper presented at the International Conference on Pervasive Computing and Communications White Plains, NY, USA.
Manzoor, A., Truong, H.-L., & Dustdar, S. (2009). *Quality aware context information aggregation system for pervasive environments.* Paper presented at the International Conference on Advanced Information Networking and Applications Workshops Bradford, UK.
Mo, Y., Chen, J., Xie, X., Luo, C., & Yang, L. T. (2014). Cloud-Based Mobile Multimedia Recommendation System With User Behavior Information. *IEEE Systems Journal, 8*(1), 184-193.
Nieto, J. F., Gutiérrez, M. B., & Lancho, B. P. (2009). *Developing home care intelligent environments: from theory to practice.* Paper presented at the International Conference on Practical Applications of Agents and Multi-Agent Systems, University of Salamanca, Spain.
Otebolaku, A. M., & Andrade, M. T. (2014). *Supporting Context-Aware Cloud-Based Media Recommendations for Smartphones.* Paper presented at the International Conference on Mobile Cloud Computing, Services, and Engineering, Oxford, UK.
Pandey, S., Voorsluys, W., Niu, S., Khandoker, A., & Buyya, R. (2012). An autonomic cloud environment for hosting ECG data analysis services. *Future Generation Computer Systems, 28*(1), 147-154.
Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *Communications Surveys & Tutorials, IEEE, 16*(1), 414-454.
Poorejbari, S., & Vahdat-Nejad, H. (2014). *An Introduction to Cloud-Based Pervasive Healthcare Systems.* Paper presented at the the 3rd International Conference on Context-Aware Systems and Applications, DUBAI, United Arab Emirates.
Poorejbari, S., Vahdat-Nejad, H., & Mansoor, W. (2017). Diabetes Patients Monitoring by Cloud Computing *Cloud Computing Systems and Applications in Healthcare* (pp. 99-116). Hershey, Pennsylvania: IGI Global.
Rahnama-ye-Moqaddam, R., & Vahdat-Nejad, H. (2015). *Designing a pervasive eye movement-based system for ALS and paralyzed patients.* Paper presented at the 5th International Conference on Computer and Knowledge Engineering (ICCKE), Ferdowsi University of Mashhad,Iran.
Ramazani, A., & Vahdat-Nejad, H. (2014). *A new context-aware approach to traffic congestion estimation.* Paper presented at the 4th International eConference on Computer and Knowledge Engineering (ICCKE), Ferdowsi University of Mashhad,Iran.
Ramazani, A., & Vahdat-Nejad, H. (2017). CANS: context-aware traffic estimation and navigation system. *IET Intelligent Transport Systems, 11*(6), 326-333.
Ranganathan, A., Al-Muhtadi, J., & Campbell, R. H. (2004). Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing, 3*(2), 62-70.
Ranganathan, A., & Campbell, R. H. (2003). An infrastructure for context-awareness based on first order logic. *Personal and ubiquitous computing, 7*(6), 353-364.
Raychoudhury, V., Cao, J., Kumar, M., & Zhang, D. (2013). Middleware for pervasive computing: A survey. *Pervasive and mobile computing, 9*(2), 177-200.
Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *Personal Communications, IEEE, 8*(4), 10-17.
Saurez, E., Gupta, H., Mayer, R., & Ramachandran, U. (2017). *Demo Abstract: Fog Computing for Improving User Application Interaction and Context Awareness.* Paper presented at the IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), Pittsburgh, PA, USA.
Shirvani, H., & Vahdat-Nejad, H. (2016). Storing shared documents that are customized by users in cloud computing. *Computing, 1*(12), 1-15.
Strang, T., & Linnhoff-Popien, C. (2004). *A context modeling survey.* Paper presented at the Workshop on Advanced Context Modeling, Reasoning and Managment as Part of UbiComp, Nottingham, UK.
Tan, J. G., Zhang, D., Wang, X., & Cheng, H. S. (2005). Enhancing semantic spaces with event-driven context interpretation *Pervasive Computing* (pp. 80-97). Berlin, Heidelberg: Springer.
Truong, H.-L., & Dustdar, S. (2009). A survey on context-aware web service systems. *International Journal of Web Information Systems, 5*(1), 5-31.
Vahdat-Nejad, H. (2014). Context-Aware Middleware: A Review *Context in Computing* (pp. 83-96). New York, NY: Springer.
Vahdat-Nejad, H., Ramazani, A., Mohammadi, T., & Mansoor, W. (2016). A survey on context-aware vehicular network applications. *Vehicular Communications, 3*, 43-57.
Vahdat-Nejad, H., Zamanifar, K., & Nematbakhsh, N. (2013). Context-aware middleware architecture for smart home environment. *International Journal of Smart Home, 7*(1), 77-86.
Vahdat–Nejad, H., Zamanifar, K., & Nematbakhsh, N. (2013). A new community–based context distribution approach for large–scale pervasive systems. *International Journal of Ad Hoc and Ubiquitous Computing, 14*(2), 90-98.
Van der Merwe, J., Ramakrishnan, K., Fairchild, M., Flavel, A., Houle, J., Lagar-Cavilla, H. A., & Mulligan, J. (2010). *Towards a ubiquitous cloud computing infrastructure.* Paper presented at the IEEE Workshop on Local and Metropolitan Area Networks Long Branch, NJ, USA, USA.

Wan, J., Zhang, D., Zhao, S., Yang, L. T., & Lloret, J. (2014). Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. *Communications Magazine, IEEE, 52*(8), 106-113.

Weiser, M., & Brown, J. S. (1997). The coming age of calm technology *Beyond calculation* (pp. 75-85). New York, NY: Springer.

Xiao, Y., Hui, P., Savolainen, P., & Ylä-Jääski, A. (2011). *CasCap: cloud-assisted context-aware power management for mobile devices.* Paper presented at the International workshop on Mobile cloud computing and services, Bethesda, Maryland, USA.

Yun, C. H., Han, H., Jung, H. S., Yeom, H. Y., & Lee, Y. W. (2009). *Intelligent management of remote facilities through a ubiquitous cloud middleware.* Paper presented at the International Conference on Cloud Computing, Bangalore, India.

Zhang, D., Cao, J., Zhou, J., & Guo, M. (2009). *Extended dempster-shafer theory in context reasoning for ubiquitous computing environments.* Paper presented at the International Conference on Computational Science and Engineering Vancouver, BC, Canada.

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications, 1*(1), 7-18.