Journal of Network and Computer Applications 40 (2014) 345–362

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Rich Mobile Applications: Genesis, taxonomy, and open issues

Saeid Abolfazli [a], Zohreh Sanaei [a], Abdullah Gani [a], Feng Xia [b,*], Laurence T. Yang [c]

[a] Mobile Cloud Computing Research Lab, Faculty of Computer Science and IT, University of Malaya, Kuala Lumpur, Malaysia
[b] School of Software, Dalian University of Technology, Dalian 116620, China
[c] Department of Computer Science, St. Francis Xavier University, Canada

ARTICLE INFO

Article history:
Received 7 February 2013
Received in revised form
1 August 2013
Accepted 18 September 2013
Available online 27 September 2013

Keywords:
Rich mobile applications
Mobile computing systems
Ubiquitous computing
Smartphone
Rich user experience
Mobile Cloud Computing

ABSTRACT

Rich Mobile Applications (RMAs) comprise a budding research area receiving increasingly abundant attention from the academic and industrial communities. RMAs are deemed to be a candidate blueprint of future online smartphone applications aiming to deliver high functionalities and rich immersive experience to mobile users. RMAs are still in early stages and comprehensive survey of the domain is lacking. In this paper, we use structuralism and functionalism paradigms to analyze RMAs' origins, trends, and characteristics. RMAs are distinguished from traditional mobile applications and Rich Internet Applications (RIAs). Comprehending the distinction between delivering Rich User eXperience (RUX) in desktop and mobile computers, and the inward similarities and dissimilarities between RMAs and RIAs will facilitate and accelerate the development of rich, smartphone-centric applications. We analyze several problems inhibiting the adoption of RMAs and review corresponding solutions to devise a taxonomy. Our study advocates that the majority of problems stem from the intrinsic characteristics of mobile devices and the heterogeneity in this environment, especially when cloud computing is employed to enhance mobile computing. Several open issues on RMAs' domination and adoption are presented as future research directions.

Contents

* Corresponding author. Tel.: +86 411 87571582.
E-mail addresses: abolfazli@ieee.org (S. Abolfazli), sanaei@ieee.org (Z. Sanaei), abdullah@um.edu.my (A. Gani), f.xia@ieee.org (F. Xia), ltyang@stfx.ca (L.T. Yang).

1084-8045/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2013.09.009

1. Introduction

Since the invention of computers, enhancing the user computing and interaction experience has been the key motivation for computing technology advancements. The invention of web in 90s significantly enhanced user experience in stationary computers, but ever increasing ownership and maintenance costs of web servers stimulated researchers to connote Rich Internet Applications (RIAs) model (Fraternali et al., 2010). The term *rich* indicates overwhelming user experience while interacting with computing devices through application — interpreted as Rich User eXperience (RUX) by the computer community, something that could not be delivered by traditional application frameworks. RIAs such as Google Map are the state-of-the-art web application model aiming to enhance desktop user experience via complex functionality, rich interaction, and compelling Graphical User Interface (GUI) while optimizing the server side costs. However, the absence of mobility in stationary computers antiquates desktop PCs and establishes a concrete ground for emergence of contemporary smartphones toward delivering RUX.

Smartphones have recently obtained momentous ground in various computing-intensive domains, particularly enterprise applications, management information systems, education, and healthcare toward surpassing desktop computers (Emmanouilidis et al., 2013; Hariharan, 2008; Martín-Campillo et al., 2013; Wu et al., 2013). Several electronic devices such as PDAs, digital cameras, Internet browsing devices, and Global Positioning Systems (GPS) are being substituted by smartphones toward all-in-one ubiquitous computing devices (http://www.gartner.com/newsroom/id/1862714). Moreover, human dependency to the contemporary smartphones is insatiably increasing in various domains due to its unique characteristics.

Success and adoption of RIAs in desktop computers and significant achievements in mobile communications and computing technologies, particularly Mobile Cloud Computing (MCC) (Sanaei et al., 2012a, 2013) beside HTML5 encourage researchers to deliver RUX to mobile users, that breed Rich Mobile Applications (RMAs). RMAs are online mobile applications that are characterized by several rich traits inherited from RIAs meant to deliver RUX to mobile users. RMAs inherit the majority of RIAs' attributes and are characterized by extensive functionality, compelling UI, high interactivity, quick responsiveness, code portability (ability to transfer application code to various platform/devices with minor/no modification Fletcher and MacDonald, 1993), and asynchronous communication — to name a few. MCC "is a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage, and mobility to serve a multitude of mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle" (Sanaei et al., 2013).

RMAs can remarkably enhance quality of mobile computing in the near future. They can significantly improve the feasibility and accuracy of several computational tasks in varied mobile environments. For instance, a mobile surgeon would be able to save a patient's life by performing remote surgery using a rich mobile surgery application. Similarly, a mobile enterprise employee can utilize GPS sensor of the smartphone to accurately and quickly locate the service delivery destination and exploit device's interactive features such as camera to record varied real-time spontaneous events during merchant delivery process. Thus, manual data-entry delay, man-made mistakes, and operation costs will substantially shrinks.

Despite the significant research about RMAs (March et al., 2011; Mukhopadhyay et al., 2009; Andruszkiewicz et al., 2011; Rahimi, 2012) and their potential advantages, there seems to be scant literature presently available. Their characteristics and technological requirements are not yet studied and the impact of smartphones shortcomings, particularly limited processing and battery capacity on RMAs are not fully explored. Hence, a need for a study that comprehensively surveys RMAs, identifies their structural and behavioral characteristics, and presents the current trends of RMAs is inevitable.

In this paper, we comprehensively survey RMAs — a first study in this area to the best of our knowledge. In order to present a clear insight into RMAs, we ascertain RMA genesis from the literature and define them using structuralism and functionalism paradigms. Structuralism and functionalism are paradigms used in social science to analyze, characterize, and interpret a phenomenon, which are extrapolated in this study. Structuralism studies the evolution of the phenomenon, compares/contrasts its structural characteristics, and unveils its limitations mostly maintaining ontology and epistemology (Burrell and Morgan, 1979). Structuralism aims to identify the underlying building blocks and their relationships to deeper comprehend the phenomenon. Functionalism, however, analyzes the current and future roles and functionalities of the phenomenon in certain environment to identify its characteristics and behavior. We employ structuralism to identify major characteristics of RMAs and synthesize them with RIAs attributes to demonstrate existing similarities and dissimilarities between RIAs and RMAs. Development problems toward RMAs' success and adoption are described and solutions that aim to alleviate current problems are taxonomized. Several open issues that grant future research directions are described. Such survey is beneficial to the mobile and networking community, because by comprehending the state-of-the-art advancements of mobile applications and computing, researchers can enhance fundamental technologies to enrich quality of future mobile systems.

The remainder of this paper is organized as follows: The genesis of RMAs is described in Section 2. Section 3 presents the definition and current trends of RMAs, and Section 4 provides a structural comparison of RMAs against RIAs. We devise taxonomy of developing problems and solutions in Section 5. Section 6 highlights open issues, and finally the paper is concluded in Section 7.

2. Genesis of RMAs

In ascertaining the origins of RMAs, we combine and extrapolate our knowledge of application architectures, chronological technology developments, and gradual evolution of major application models, namely (i) standalone applications, (ii) client–server applications, (iii) web applications, and (iv) RIAs. For analysis, we opted for five metrics, namely application architecture, computing distribution, application portability, offline usability, and user experience — all influential factors for enhancing application functionality and the quality of user experience. Studying the genesis of RMAs is beneficial to comprehending conceptual foundations, vision, and hype. Application evolution is explained next, and the results of our analysis are presented in Fig. 1.

Standalone applications are monolithic platform-dependent applications that perform entirely on a single processing unit and reflect to the users' actions based on the hosting machine's computing speed. Such single-tier architecture demands rich native computing resources and limits code execution performance to the computing capabilities of the hosting machines. They cannot perform outsourcing and load balancing mechanisms in case of excessive computing load. Moreover, the applications' code and data are vulnerable to several threats like physical damage, hardware malfunction, and robbery of hosting machine. However, standalone applications are protected against external threats in the absence of intersystem communications. Monolithic applications on the hosting machines must be installed and maintained, and every code update should be applied to all available copies. Due to various types of heterogeneity in client side (e.g., hardware, Operating System (OS), and programming language), various versions of a single application should be developed for different platforms, a practice that is a costly and time consuming (Sanaei et al., 2013). Such limitations beside the exponential growth in data volume and processing tasks, inefficiency of terminal-mainframe communications, and noticeable advancements in networking technologies, paved the way in the 1970s for innovation of the client–server model (Duo et al., 2002).

Client–server applications are applications whose functionalities are logically separated among more than one computing entity, e.g., client and server. Thus, they are named multi-tier applications. The client–server model enables distribution of computation-intensive tasks by leveraging resource sharing and remote execution mechanisms, which in turn facilitates execution of computing- and data-intensive processes. In client–server application model, heavy and resource-intensive components of the application are executed in the powerful server and lightweight components are running inside the client device. The client initiates the application and calls for remote execution when required. However, client-side code portability, programming complexity (due to intricate intersystem networking operations), and necessity to update client codes are challenging tasks that inhibit client–server applications' adoption.

To alleviate these challenges and provide ubiquitous access to the code and data, web applications were invented in the convergence of client–server architecture and hypertext (Duo et al., 2002). Web

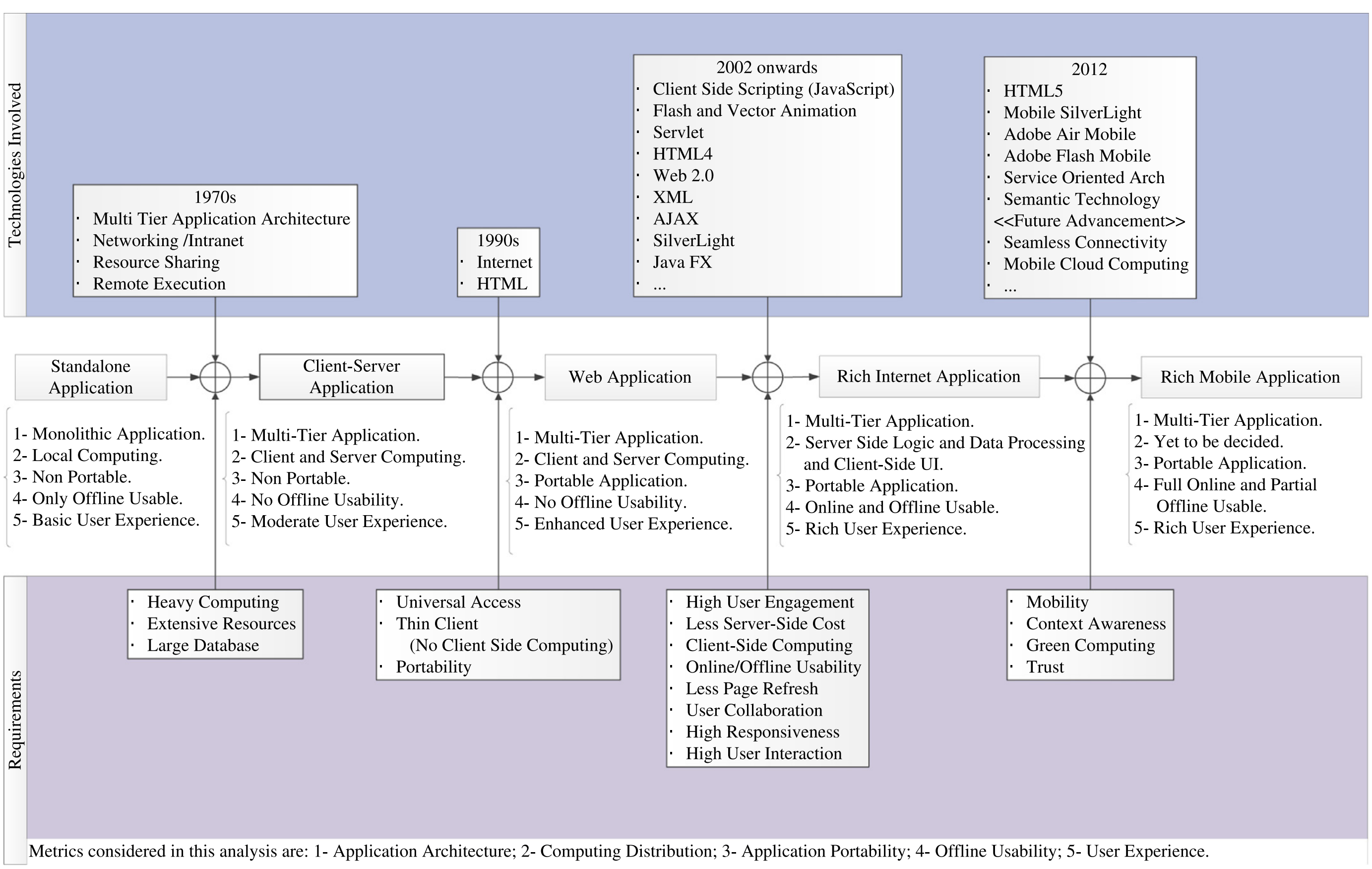

Fig. 1. *Genesis of Rich Mobile Applications* – This figure shows gradual development of candidate application architectures from early desktops till recent years from left to right that ends with RMA. Five metrics are studied for candidate architectures and results are summarized in the central segment. At the bottom of the diagram, the requirements of every era are briefed and the top segment shows involved technologies. Note that this figure shows a narrow field of technology and architecture development which is related to RMA.

applications are multi-tier, browser-dependent, portable applications run on various computing devices regardless of underlying platform. They deliver ubiquitous access to functionality and data and utilize the channel of Internet or Intranet for intersystem communications. Any update in the application code is performed inside the server machine so the clients can access the latest version without updating effort. However, web application development, maintenance, and management became extremely complex, because such huge number of transactions needed human interpretation and intervention in the absence of machine-comprehensible transactions. Sharp growing of inscrutable information, synchronous communication, and lack of user contribution and social interaction in the web applications (Cormode and Krishnamurthy, 2008) intensify such complexity and outpaced web technology of the time. In 1999 DiNucci coined the term *Web 2.0* as the future web technology. Thereafter, web 2.0 bloomed to address several insufficiencies of traditional web (i.e., web 1.0) to deliver fascinating, engaging online experience to end-users.

Although web 2.0 could deliver platform-independent, rich, online computing experience to the end-users, their intensive back-end processing drastically raised the ownership and maintenance costs of servers. To mitigate such cost and complexity and stimulate green computing, researchers utilize the computation power of PCs that are able to perform heavy computing in client side. They separate presentation layer from the application and transferred UI (User Interface) to the client side to lessen server-side costs and complexities, reduce communication overhead, and enhance the application responsiveness which bred RIAs (Christodoulou et al., 1998; Duhl, 2003). RIAs are multi-tier applications that converge the lightweight, distributed architecture of web applications with aesthetically pleasing, interactive, and easy-to-navigate interface of desktop applications to deliver RUX to stationary users. RIAs employ technologies such as HTML5, XML, and AJAX (Meliá et al., 2010; Deb et al., 2007; Noda and Helwig, 2005) for delivering portability, online/offline functionality, and ubiquitous data access via persuasive UI, rich interaction, and crisp response. Because presentation layer in RIAs is migrated to their immediate environment, they can benefit from context- and social-aware computing.

Parallel to the RIAs' success in stationary computers and advancements in ubiquitous computing (West, 2011), smartphones have gained noticeable attention because of their unique capabilities and features, particularly mobility, context awareness, context sensing, multimedia, and location-based services such as Kim (2011a). Researchers endeavored to empower smartphones functionality, enhance user experience, and realize computing richness in mobile devices that bred RMAs. NTT DOCOMO used Adobe Flash Lite for the first time in 2003 to increase mobile applications' functionality and engagement. Later in 2008, Google shifted 'Google Gear' to Windows Mobile 5 and 6 devices to support platform-neutral mobile applications in offline mode. Google Gears for mobile devices is a mobile browser extension for building rich web applications. Such web applications can be executed inside the mobile browser in every mobile device with a web browser. Later in April 2008, Microsoft released Silverlight for mobile platforms to develop engaging, interactive UIs. Silverlight is a .NET plug-in compatible with several mobile web browsers that can execute the Silverlight-enabled mobile applications. In September 2008, Android incorporated the Google Gear plug-in to the Chrome Lite browser to enrich the interaction experience of Android end-users. Thenceforth, richness dramatically entered into the mobile domain and encouraged rich applications in users' pockets.

3. RMAs: definition and current trends

In this section, RMAs are defined from structural and functional points of view according to the observation from current research & development efforts and the essence of what RMAs are promising to deliver. In addition, we explain emerging trends of mobile applications compared to desktop applications.

3.1. Definition

Rich Mobile Applications are energy efficient, multi-tier, online mobile applications originated from the convergence of mobile cloud computing, future web, and imminent communication technologies envisioning to deliver rich user experience via high functionality, immersive interaction, and crisp response in a trustworthy wireless environment while enabling context-awareness, offline usability, portability, and data ubiquity.

At a superficial glimpse, RMAs are suitable for various computing devices like smartphones, PDAs, and tablets, but the recent emerging trend of smartphones (http://www.gartner.com/newsroom/id/2408515) dedicates RMAs to smartphones because PDA popularity is sharply fading (BCC Research, 2009) and slow-growing tablets (190 million tablet shipping against 1.9 billion smartphone, http://www.gartner.com/newsroom/id/2408515) are more like laptops than mobile devices. In addition, PDAs, tablets, and laptops cannot acquire sufficient knowledge from the surrounding environment due to limited sensing ability that inhibits context awareness in RMAs. RMAs' execution is initiated from the hosting mobile devices; therefore, they are different from web applications that start up remotely in a back-end machine.

3.2. Current trends

Figure 2 displays the popularity trend of mobile versus desktop applications from 2004 till present, as measured by Google Trend. The constant trend of desktop applications versus the drastically hiking trend of mobile applications advocates a demand for mobile applications. The prominent spot points in Fig. 2 indicate that mobile applications usability is sharply increasing in several domains including business, government, health, and social networking. Spot points related to mobile application are as follows:

(A) WebMD Launches New Version of Free Mobile Application for Physicians CNNMoney.com — February 16, 2010
(B) LexisNexis(R) Launches Accurint(R) Mobile Application for Government and LawNewsday — July 12, 2011
(C) Health Insurance Launches Mobile Application to Help Consumers Compare and Apply for Health Insurance Coverage Market Watch — September 19, 2011
(D) Facebook acquires mobile application firm Gowalla Economic Times — December 6, 2011

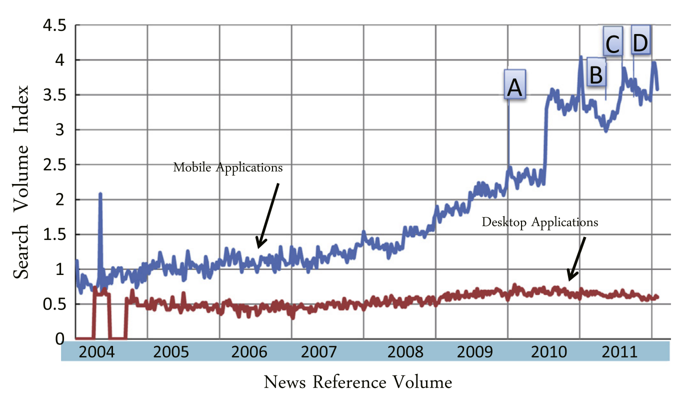

Fig. 2. Google search trend for mobile and desktop applications since 2004

Table 1
Comparison of rich characteristics between RIAs and RMAs.

Metrics	Rich Internet Applications	Rich Mobile Applications
Functionality	High	High
User interface	Rich and complex with comprehensive details	Rich, simple, and adaptive with short detail due to space limit
Interactivity	Medium due to limited interaction tools (e.g., mouse and keyboard)	High due to variety of interaction tools (e.g., touch screen, camera, several sensors)
Context-awareness	Low due to lack of context gathering sensors	High due to mobility and wide range of sensors
Trustworthy	Medium due to wired connectivity and Internet	Low due to wireless connectivity, Internet and mobility
Energy efficiency	Less critical due to uninterruptible power source	More critical due to limited battery source
Responsiveness	High due to rich resources and wired connection	Medium due to limited resources and wireless connection
Portability	High due to web as underlying platform	High in web-based, low in others due to heterogeneity
Online connectivity	High due to wired connection	Medium due to wireless connection
Offline usability	High due to high bandwidth and large storage	Medium due to low bandwidth and limited storage

Moreover, Gartner has identified mobile applications as the new, key strategic technology trend of 2011. Meanwhile, technology leaders have been increasingly collaborating to convey RUX to mobile users. For example, Microsoft Silverlight Mobile is a platform to conveniently develop and integrate rich interactive features (e.g., pixel shader and sketchflow) to resource-constrain mobile devices regardless of time, place, or platform. Oracle Sun has introduced Java FX Mobile technology to actuate generating cross-platform mobile applications adaptive to all screen types, e.g., mobile, tablet, laptop, and TV. By using Adobe Air, developers can build off-browser standalone applications for PCs and mobile phones. Flash Builder and Flex of Adobe enable developers to use a common codebase to build cross-platform applications for Android, Blackberry, tablet OS, and iOS. Leading companies like IBM and HP are also contributing in R&D activities. The efforts of IBM and HP research groups are to enhance user experience and pave the way toward RMAs development. IBM Mobile Portal Accelerator is a content/application adaptation product that provides rich mobile-centric experience to users of various mobile devices with different visualization capabilities. µCloud (March et al., 2011) is a recent cloud-based RMA framework developed by HP that aims to facilitate RMA development using cloud resources. Such innovations and contributions from the academic and industrial communities advocate rapidly emerging trend of RMAs in the near future.

4. RMAs versus RIAs

Although RMAs inherit their properties from RIAs, their similarities are mostly ostensive due to intrinsic smartphone properties and the mobile environment. In this section, the major characteristics of RMAs are described based on literature and current smartphone advancements as well as application and user requirements. Several differences in the way richness is delivered to immobile and mobile users are discussed and summarized in Table 1.

4.1. RMAs characteristics

RMAs are aimed to provide extensive mobile computing and comprise a suitable ground for a plethora of domains such as healthcare, emergency handling, disaster management, crowd management, urban and rural development, the tourism industry, and multimedia. Using an imminent RMA, a surgeon would be able to perform remote operation to save a patient's life using an application developed based on RMA characteristics. Therefore, they are characterized to meet computing requirement of all mobile service consumers, including individual, corporate, and enterprise users. Although richness is the common attribute in RMAs, its degree varies among varied applications depending on how rich traits are combined and deployed in different RMAs. Traditional application models demand a high degree of user concentration and competence, and consume noticeable amounts of resources that are not appropriate in such domains.

Richness among RMAs differs due to several influential metrics such as (i) appropriate composition of technologies and characteristics, (ii) applying user preferences and expectations in the design level, and (iii) the quality of execution environment (including quality of wireless communication and computational capabilities of hosting device). Certainly, basic characteristics of software systems and applications (that are not discussed here) such as consistency, reliability, and scalability (Birman, 1997; Bondi, 2000) are necessary for the success of RMAs. Several key attributes of RMAs that are realized or will be realized in near future are explained as follows.

4.1.1. Rich functionality

RMAs deliver overwhelming functionality compared to traditional mobile applications, regardless of smartphone resource constraints. From user perspective, rich functionality is the ability to conveniently perform any computation regardless of the application resource requirements and mobile device's constraints. Rich functionality is directly affected by the architecture, logic, algorithm, programming language, and computing technologies used in design and development phases. For instance, a highly functional navigation application utilizes available features of the hosting device (e.g., GPS, accelerometer, gyro, and multimedia) to minimize user effort (e.g., automatically identifies current location and driving speed) in fulfilling users' needs. Maintaining adequate map including a large POI (Place of Interest) database in addition to a lightweight, optimum, consistent search algorithm can deliver the essence of rich functionality to the users navigating on the go. However, rich functionality increases the development and execution complexity, imposing extra effort on the application analyst, designer, and developer, and requires several cornerstone technologies.

4.1.2. Rich user interface

One of the pivotal traits of RMAs is to deliver rich and compelling UI allowing end-users to effectively interact with the device, perform computing and communication transactions, and control and customize the device according to their preferences and requirements. According to the user-centric design UI process (Norman and Draper, 1986), rich interaction experience can be achieved by rich UI design and intuitive interaction that are explained as follows.

Rich design: Rich and user-centric mobile interface leveraging modern interaction technologies, particularly touch, gesture, multimedia processing, and sonification (conveying a message or alert to user via non-speech audio, Kramer, 1994) is transforming traditional

UI to rich UI (Ferguson et al., 2012). Microsoft Silverlight, Adobe Air, and Adobe Flash are eminent industrial efforts to enrich mobile user interface using various multimedia and visual effects. In an attempt to create guidelines for designing a rich UI — taking into account the ergonomic principles — IBM-acquired Worklight Inc. (http://worklight.com/resources/webinar-designing-rich-mobile-apps) identifies three comfort zones, namely Easy, Reach, and Medium (Fig. 3). Positioning touchable icons in the Reach and Medium zones likely degrade Quality of Experience (QoE) — since accessing objects in these areas is not easy for the majority of users — whereas utilizing the Easy zone is convenient for a wide range of users to tap. However, the enlarged screen of contemporary smartphones changes the way users interact with devices. Though they are handheld devices, it is awkward to hold with a single hand and interact with them.

Easy and less astonishing: Intuitive, simple, well-designed UI is one of the major factors contributing to the astounding success of RMAs. According to the principle of least astonishment, applications should look and behave analogous to users' mental expectations — not according to the application's functionalities — in order to keep them engaged and satisfied (Bloch, 2006). However, designing a simple and less astonishing UI, especially in a small visualization area of a mobile device is a non-trivial task. Very large number of + mobile users are experiencing digital interaction for the first time. Besides, most of them are in a hurry and have less concentration while interacting with the device. Hence, a rich UI can effectively enable end-users to interact with the device to obtain the required output with the least concentration while still are able to undo possible mistakes and correct them.

4.1.3. Immersive interactivity

Smartphones are an intrinsically suitable platform that leverages expressive interaction and conveys RUX to mobile users by responding to their instantaneous reflections and gestures. At first glimpse, tiny physical or virtual keyboards and lack of mouse seem to be serious problems, but mobile applications are privileged to alleviate contemporary digital features, especially tilting sensors, accelerometer, GPS,

Fig. 3. Screen comfort Zone of Rich UI (Webinar: Designing RMAs in a Fragmented World, 2011).

and camera to offer alternative interaction techniques. Moving a magnet near smartphone (Yuksel et al., 2011), using accelerometer for data entry (Valberg and Christensen, 2009; Jones et al., 2010), tilting handset to browse photos (Cho et al., 2007), and tapping the screen to mute and unmute the sound (Eslambolchilar and Murray-smith, 2004) are among the alternative smartphone interaction methods. The smart composition of these interaction methods for designing multi-modal mobile applications (Anegg et al., 2004) is vital in delivering RUX to mobile users.

4.1.4. Context-awareness

RMAs can leverage context information to deliver user-centric functionality, UI, and interaction to mobile users. Context is a piece of information that can be used to characterize the structure and behavior of various entities such as person, place, environment, or object (Dey, 2001). For instance, in mobile computing, context can be geographical coordination of a user, weather humidity, bandwidth availability, mobility traits, and user preferences which are not difficult to acquire with current smartphones (Conti et al., 2012). Smartphones utilize multiple tools and technologies such as accelerometer, light sensor, compass, and GPS to acquire comprehensive knowledge about their surrounding environment and people (Lane et al., 2010). Context in RMAs can be utilized to adapt content and presentation of functions and services according to individual users with the aim of generating user-centric context-aware applications (Kaasinen, 2003).

4.1.5. Trustworthy

Enhancing security and privacy provision in resource-constraint mobile devices tranquilize users and enrich their mobile computing experience, but decrease computing performance due to extra security-related computations and controlling overhead (Almenares et al., 2013). Security and privacy are crucial for smartphone users from several points of views, including location information, online behavior, and stored data (Prosper Mobile Insights, 2011) at a time when mobile hacking and malware-contaminated applications are on the drastic rise (Cachin and Schunter, 2011; La Polla et al., 2012). Several security threats have been identified in an experiment of Android mobile applications (Enck et al., 2011) with the potential to violate the security of mobile users. Moreover, contemporary features, particularly GPS and accelerometer can threaten users by capturing their location and credential data such as passwords (Marquardt et al., 2011). Performing complicated resource-intensive security algorithms is infeasible due to resource deficiencies of mobile devices. Therefore, it is crucial to assure mobile users, especially corporate business users, regarding the security and privacy of their data.

4.1.6. Energy efficiency

Energy efficiency is an inseparable trait of RMAs because a battery-hungry application can quickly deplete available energy source and hinder the usability of applications main functionality and secondary rich traits. The miniature nature of mobile devices and user safety concerns inhibit exploiting large battery pack with mobile devices and frequent recharging battery is one of the most aggravating impediments for mobile users (Sanaei et al., 2012b). Thus it is vital to deliver complex high-performance functionalities at the cost of the least energy demand. Researchers have been endeavoring to enhance energy efficiency in mobile devices using varied approaches, particularly energy-aware hardware and OSs, fidelity adaptation, efficient resource management, and Mobile Cloud Computing (MCC) solutions (Abolfazli et al., 2012a; Bianzino et al., 2012; Vallina-Rodriguez and Crowcroft, 2013). In MCC, researchers exploit cloud-based resources to augment computational capabilities of mobile devices to not only conserve local

Table 2
Structural comparison of RIAs and RMAs.

Characteristics	Rich Internet Applications	Rich Mobile Applications
Architecture	Multi-tier, web application	Multi-tier, online smartphone-nested applications
Task distribution	Presentation layer (GUI processing, multimedia execution, and request initiation) and partial database run inside desktop machine and rest remotely in the Web	Native codes as well as partial UI and database run on mobile devices, rest remotely on servers
Execution environment	Browser-dependent and browser-independent	
Interaction Tools	Mouse, keyboard, (touch) screen, voice and camera	KB, touch screen, voice, camera, and context acquisition sensors
Interaction Limit	Mostly no touch screen, camera. no sensing tools	No pointing device, small screen
Screen Layout	Large and landscape,	Small and mostly portrait
Usability	Online and offline	Intermittent online and limited offline
Start-up time	High due to remote startup	Low due to local startup
Native feature dependency	Light dependency to features like microphone, camera, finger scanner	Strong dependency to features like camera, sensors to enhance interaction and user experience
Portability	High in browser-based applications low in standalone applications	
Communication technology	Asynchronous communication	
Network medium	Mainly wired	Wireless

resources, particularly battery, but also alleviate rich functionality by execution of substantially compute-intensive applications on the go (Abolfazli et al., 2013).

4.1.7. Crisp response

Spontaneous reflection to changes/actions and arrival of data are desirable to users and hence, are crucial for RMAs. Application processing time and communication latency are major evaluating metrics to convey a sense of responsiveness to users. Though tolerable waiting time for web users is recently measured as 2 s (Forrester Research, 2009), it is an unacceptable delay to thin-client applications (Tolia et al., 2006); smartphones, as thin computing devices, strive for crisp response of less than 150 ms. However, response delays of less than 1 s are noticeable to annoying, while delays of more than 1 s pose potential threats to user experience and should be avoided.

4.1.8. Cross-platform cross-device

Developing cross-platform and cross-device RMA significantly enriches developers and end-users experience. Enterprise employees are not interested in corporate-issued smartphones and prefer to use their personal devices for business purposes (http://infoworld.com/d/mobilize/mobile-management-infoworlds-expert-guide-371-0). In the absence of portability trait of mobile applications, developers should write different versions for a single mobile application, each for a specific platform — this not only imposes financial costs, but also significantly increases development time (Sanaei et al., 2013). Although realizing portability for RMAs is a challenging task, it significantly enhances quality of mobile users' experience.

4.1.9. Ubiquitous functionality and data access

Multi-tier architecture of RMAs enables distribution of task between native device and remote servers resulting in seamless ubiquitous functionality and data access which can realize the true feel of real time well-connected mobile applications to mobile users. Responding to asynchronous events as they occur in smartphones and back-end servers is an intrinsic trait of RMAs. In the event of network failure, techniques like widgets, push messaging, and partial cached data are suitable just-in-time reactions that convey a sense of connectivity to users. If developers assume that mobile devices are always connected to the web, application usefulness increases and user experience is enhanced. Because mobile users confront various connectivity levels at different times (i.e., high/low bandwidth and limited or no connectivity, Oehlman and Blanc, 2011), it is important to consider heterogeneous connectivity and bandwidth to provision offline support while designing and developing RMAs. However, ubiquitous content access while client, content, or network is moving demands forthcoming technologies and efforts (Yu et al., 2012; Ahlgren et al., 2012).

4.1.10. Offline usability

Ideally, RMAs should deliver full functionality and content delivery in both online and offline modes to noticeably improve user satisfaction. Thus, users would still be able to read and reply to emails, even though the device is disconnected from the communication network. The changes in offline mode will be synchronized when the user returns online. RMAs can deliver offline usability via caching, state management, and data access mechanisms that requires large native storage as a non-trivial issue (Meier et al., 2009). Google presently provides offline Gmail facilities to their smartphone clients using browser, which is a step toward delivering richness to end-users (King, 2013).

4.2. Structural comparison of RMAs and RIAs

While the common vision of RIAs and RMAs is to deliver RUX, there are several structural discrepancies in fulfilling such aim. Table 2 presents major characteristics of RIAs and RMAs that are briefly analyzed and synthesized as follows.

While both RIAs and RMAs are multi-tier online applications, in RIAs the logic and data layers are located in the back-end server and the presentation layer is migrated to the client machine. The main principle in RIAs is to utilize the local computing resources of desktop machines because PCs have giant, multi-core processing resources with vast memory and storage able to process medium-intensive tasks such as rendering UI, manipulating data, and utilizing multimedia content without connecting to a server. Local processing (including partial database storage) in resource-rich, contemporary PCs reduces client–server networking traffic and delay, and shrinks ownership and maintenance costs on the server side. However, this is not applicable to RMAs. One probable distribution model — in the absence of widely acceptable architecture for RMAs — is to keep a partial copy of the application (including the presentation layer, logic, and data layer) in the

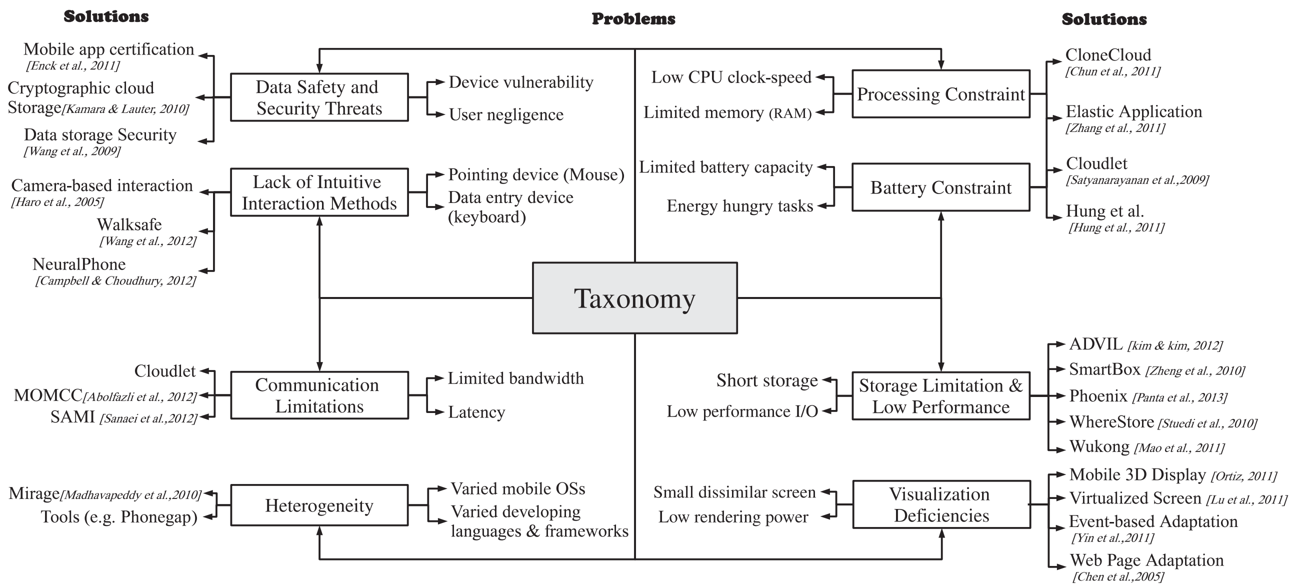

Fig. 4. Taxonomy of RMA development problems and solutions

mobile device and distribute the remaining components (which are likely resource-intensive tasks and contents) among private servers or public resources, particularly clouds. However, native components (e.g., camera-related codes) of the mobile application cannot be migrated from the device; non-native codes are free to be bounded to the smartphone or remote server.

From a communication perspective, both RIAs and RMAs leverage asynchronous communication mechanism as the most predominant technology for enhancing the quality of communication in online applications (Campesato and Nilson, 2011). In asynchronous communication, the user request will be instantly forwarded to the server (either local or remote server) in the background while interaction can continue without interruption till the request is served and response is received. Eliminating redundant messaging between client and server reduces communication latency, something more crucial in RMAs due to the characteristics of the wireless medium.

In terms of application usability, offering online and offline application functionality and access to content in RIAs and RMAs is necessary to fulfill user requirements and deliver RUX. In RIAs, desktop computers and LAN-connected laptops are well connected to Internet via broadband, leveraging high bandwidth and low jitter and latency. Such reliable communication mediums (compared with the wireless medium) and giant disk storage with terabyte spaces, facilitate online and offline functionalities and content delivery in immobile computers. However, wireless communication mediums of smartphones are comparatively low-bandwidth, error-prone, and intermittent with high jitter and long WAN latency. Limited smartphone disk storage in the wireless environment hampers online usability and offline functionality of applications. Disconnecting from the network impedes application usability in the absence of offline functionality and local data storage.

Visualization capabilities and interaction mediums are additional differentiating aspects of RIAs and RMAs. Desktop users benefit from large screens, full-size physical keyboards, and multi-key mice. However, a small mobile screen and the lack of a physical keyboard or mouse differ the interaction quality of mobile applications from desktop applications. Data entry and interaction using contemporary mobile devices have recently revolutionized utilizing modern data acquisition tools like accelerometer, compass, GPS, camera, and thermal sensors tools to enhance user experience (Lane et al., 2010; Koukoumidis et al., 2012; Song et al., 2010). However, sensing large amounts of data consumes local mobile device resources. Storing and interpreting such extensive data are also challenging tasks that require future research and development.

5. Taxonomy of RMA development problems

RMAs are likely a future generation of smartphone applications with the likelihood of becoming dominantly popular as compared to other mobile and desktop application models such as RIAs or web applications. However, several problems encumber RMAs' adoption and success. In this section, we discuss RMAs' development problem and review efforts and proposals that aim to alleviate these problems. The devised taxonomy is depicted in Fig. 4.

5.1. Processing constraint

Processing deficiencies of mobile clients due to low CPU speed and limited RAM is one of the major problems in mobile computing (Abolfazli et al., 2012a, 2013). Users constantly envision using smartphones with similar computing capabilities to desktop machines to perform heavy computing tasks on the go. Such vision requires rich resources, including a powerful processor and large memory. Although smartphone processing abilities are on the increase, user expectations — especially business users — are still far beyond the actual smartphones' processing capabilities.

To alleviate processing limitations of smartphones, manufacturers are endeavoring to produce multi-core high clock-speed CPUs and large memory. However, it is very slow and costly solution considering the current knowledge and technologies that drastically increases mobile devices' prices and makes them less affordable. Alternatively, researchers aim to conserve local smartphone resources by computation augmentation. Mobile computation augmentation “is the process of increasing, enhancing, and optimizing computing capabilities of mobile devices by leveraging varied feasible approaches, hardware and software.” (Abolfazli et al., 2013). Leveraging mobile augmentation solutions, RMAs can provide more complex functionality to the users, consume less energy, and perform tasks in less time — if augmentation overheads do not exceed conserved resources (Sharifi et al., 2011). While efforts for manufacturing high-end hardware devices are in progress, software approaches are tremendously emerging. MCC solutions are the most prominent software approaches to alleviate smartphone shortcomings.

In the state-of-the-art MCC efforts, researchers (Satyanarayanan et al., 2009; Chun and Maniatis, 2010; Kemp et al., 2012; Zhang et al., 2011a; Cuervo et al., 2010; Chun et al., 2011; Verbelen et al., 2012; Broberg et al., 2009; Abolfazli et al., 2012b; Sanaei et al., 2012a) exploit four types of cloud-based computational resources, namely immobile distant clouds (e.g., public clouds like Amazon EC2), proximate immobile computing entities (e.g., PC in coffee shops), proximate mobile computing entities (e.g., smartphones), and hybrid (e.g., Amazon EC2 and PC in coffee shops) (Abolfazli et al., 2013). Table 3 summarizes the characteristics of these four types of cloud-based resources that are presented in our recent study (Abolfazli et al., 2013). Some of the most credible cloud-based augmentation efforts in MCC that use such heterogeneous cloud-based resources for augmentation are reviewed as follows.

CloneCloud (Chun et al., 2011) uses immobile distant cloud resources. It is an application partitioner and execution runtime framework that clones the entire mobile platform into the cloud Virtual Machine (VM) and runs the mobile application inside the VM without performing any change in the application code. The CloneCloud enables partial execution of mobile application locally when the intensive components are running in the cloud unless local execution tries to accessing the shared memory state. CloneCloud can substantially decrease the overall execution time via thread-level code offloading. When the native execution reaches the intensive component(s), the CloneCloud system migrates the computations to the cloud and continues native execution until the application fetches data from the offloaded state. The native execution is paused until the results are returned and integrated to the local application. However, the virtualization overhead, clone transferring cost to the cloud, and recurring synchronization of the shared data between the mobile and cloud degrade augmentation performance (Shiraz et al., 2012c).

Elastic application model (Zhang et al., 2011a) employs immobile distant clouds to run resource-intensive components of RMAs. In this work, mobile application is partitioned into several loosely coupled, small components, named Weblet. The Weblet execution is configured at runtime to either run locally or remotely, based on resource intensity of Weblets, quality of runtime environment, and offloading goals. RMAs' execution can benefit from parallelism feature of this proposal, because every Weblet is being sent to more than one remote resource and results can be integrated into the native components of the mobile device. Parallel execution of intensive components can improve augmentation robustness in scenarios where a remote code terminates its execution service. In this solution, local versus remote execution configurations is specified at runtime to match the application requirements and to enrich user experience. A particular Weblet can be remotely executed in a low-end computing entity while the same Weblet can be locally executed on a high-end smartphone. Elastic application model enriches user experience by enabling different running modes of a single application (e.g., high speed, low cost, and offline mode). However, it engages application developers to determine weblets organization based on the functionality, resource requirements, and data dependency.

Researchers in Satyanarayanan et al. (2009) use proximate immobile computing entities and propose a VM-based mobile augmentation solution that exploits Cloudlet resources to augment neighboring mobile devices with least security risks and communication latency. Cloudlet is remote computational resources composed of one/multiple resource-rich, multi-core computing entity (ies) connected to uninterruptible power and high speed network link. Mobile device plays the role of a thin client while the intensive computation is entirely migrated via IEEE 802.11 to the Cloudlet in vicinity. It maintains distant clouds as backup resources in case of Cloudlet scarcity. Cloudlet assumes that the complete clone of mobile OS is loaded in the remote computing devices running on an isolated VM prior the augmentation process. In mobile side, unlike CloneCloud that clones the entire OS Cloudlet leverages VM overlay which is a lightweight software interface of the intensive components. The authors employ Dynamic VM Synthesis (DVMS) method assuming that the base VM is already available in the target Cloudlet and user can find the matching execution environment (VM base) among silo of nearby Cloudlets. Cloudlet's novelty comes from the compact size of VM overlay that significantly shrinks the communication overhead, reduces application execution time, and saves energy. Upon successful negotiation of the system with nearby Cloudlet(s), the DVMS migrates the VM overlay and executes launch VM (base+overlay) inside the Cloudlet. Hereafter, the Cloudlet executes remote components, sends back results and integrates them to the mobile device. Despite the noticeable improvements, Cloudlet

Table 3
Comparison of cloud-based resources for mobile computation augmentation.

Metrics	Distant immobile cloud	Proximate immobile computing entities	Proximate mobile computing entities	Hybrid
Architecture	Distributed			
Ownership	Service provider	Public	Individual	Hybrid
Environment	Vendor premise	Business center	Urban area	Hybrid
Availability	High	Medium	Medium	High
Scalability	High	Medium	Medium	High
Sensing capabilities	Medium	Low	High	High
Utilization cost	Pay-as-you-use			
Computing heterogeneity	High	Medium	High	High
Computing flexibility	High	Medium	High	High
Power efficiency	High	Medium	Medium	High
Execution performance	High	Medium	Medium	High
Security and trust	High	Moderate	Low	High
Utilization rate	High			
Execution platform	VM	VM	Physical/VM	Physical/VM
Resource intensity	High	Moderate	Moderate	Rich
Complexity	Low	Moderate	Moderate	High
Communication technology	3G/Wi-Fi	Wi-Fi	Wi-Fi	3G/Wi-Fi
Mobility	NA	NA	High	High
Communication latency	High	Low	Low	Moderate
Execution latency	Low	Medium	Medium	Low
Maintenance complexity	Low	Medium	Medium	High

requires plethora of powerful Cloudlets containing popular mobile platforms' base VM which is not always feasible. Incentivizing and providing adequate security provisions to encourage individual computer owners to deploy Cloudlets must be addressed before successful deployment of Cloudlet in practice. Moreover, further efforts are required to enhance the overall offloading process in terms of energy efficiency, application interactivity, security, privacy, and maintenance of Cloudlet to ensure application responsiveness and rich user experience.

Hung et al. (2012) proposes a mobile augmentation framework that offloads execution of intensive Android mobile applications to immobile distant cloud's virtual execution environment. The efficiency and precision of runtime environment is noticeably impacted by the completeness and fidelity of emulated platform. The authors exploit software agents in smartphone and cloud counterparts to decrease system complexity and facilitate its management and maintenance. The agent in mobile side creates VM and clones the application in whole (even native codes and UI components) with partial data/memory state and migrates to the cloud. Different from CloneCloud, this proposal reduces latency by partial offloading part of the data stack that is explicitly created and solely owned by the application to the VM instead of cloning the complete memory. In remote execution stage, the system continuously updates the progress between smartphone and cloud to ensure integrity of both copies. Probabilistic QoS-aware communication technique in this effort aimed to perform a communication-QoS trade-off to satisfy the requirements and enrich user experience. For example, the control data which are often small in size can be given a reliable communication technique, whereas voluminous video streaming data can be transmitted via less accurate communication method. Peripheral features in this solution such as automatic virus scanning, data imaging and recovery, and data sharing in the virtual environment are promising to enhance quality of user experience. In this effort, RMA's crisp response and rich interactivity are degraded due to the remote UI execution. To enhance responsiveness and interactivity of RMAs, UI and interaction components can be executed locally instead of migrating the entire application to the cloud.

SAMI (Service-based Arbitrated Multi-tier Infrastructure for mobile cloud computing) (Sanaei et al., 2012a) proposes a novel cloud-based computation augmentation solution that leverages a 3-tier cloud-based resources, namely public clouds, Mobile Network Operators (MNO) as one-hop cloud provider for mobile users, and authorized MNO dealers that are distributed in urban and rural areas. In this solution, two types of cloud-based resources are utilized in its hybrid resource type, which are distant immobile clouds and proximate immobile computing entities. The main rational behind such hybrid infrastructure is to provide highly elastic localized computational resources anywhere anytime. In this model heterogeneous communication technologies, namely IEEE 802.11 and cellular technologies are used; IEEE 802.11 enables energy efficient communication with nearby MNO dealers since it consumes less energy than cellular (Perrucci et al., 2011), whereas cellular technology (e.g., 2G and 3G) realizes communication with public clouds and MNOs. Although SAMI is a conceptual framework, its hybrid solution can significantly reduce long WAN latency and increase energy efficiency for execution of RMAs toward RUX. Moreover, authors need to discuss a detailed discussion of a feasible business plan for SAMI.

5.2. Battery constraint

Battery power is the only non-replenishable resource in smartphones that requires external resources to be renewed (Miettinen and Nurminen, 2010). Current smartphones are equipped with a lithium-ion battery with less than 3000 mAh capacity that lasts few hours when they are used as computing or communicating devices. Manufacturers' ideals of attaining device lightness, compactness, and handiness are preventing the production of bulky batteries, unless alternative power technologies such as nuclear energy that can last month or years imminently enter the market. Moreover, user safety is another vital issue that restrains hardware producers not to produce high capacity, long lasting batteries (Ben et al., 2009). Explosion of a battery with few hundreds milliamperes capacity is potentially dangerous and can threaten users' health.

Additionally, battery capacity growth is as slow as about 5–10% per annum (Neuvo, 2004) because current battery cells are excessively dense (Satyanarayanan, 2005). Via sundry energy harvesting efforts (Flinn and Satyanarayanan, 1999; Starner et al., 1997) researchers are endeavoring to restore battery power from renewable resources such as muscle power, wireless signals, and solar power. Nevertheless, usability of these resources is inhibited due to their unreliability and intermittency (Pickard and Abbott, 2012). For example, an immobile end-user at dark cannot replenish his/her device battery in the absence of uninterruptible wall power and wireless radiations. Recent researches are toward developing alternative long-lasting energy sources, particularly nuclear batteries that can last for months or years (Wacharasindhu et al., 2009). However, future R&D efforts are essential to fulfill insatiable energy demands of mobile users.

The energy deficiency of mobile devices is further intensified by energy-hungry native transactions. The majority of processing tasks, particularly I/O transactions and wireless communications are identified as energy-hungry components in smartphones that can sharply deplete native energy. For instance, every 1 MB of data storage/retrieval consumes about 600 mW of energy (Perrucci et al., 2011). Though energy-aware algorithms and context-aware selection of communication medium from a pool of heterogeneous technologies can conserve mobile batteries (under investigation in the next generation of wireless systems, Akyildiz et al., 2004; Kim and Mohanty, 2010), the energy resource limitations in smartphones remain a problematic challenge in the development of RMAs.

Researchers efforts in conserving mobile energy can be classified into four major categories of (i) developing energy-aware OSs, (ii) optimizing wireless communication interface and protocols, (iii) optimizing sensors, and (iv) augmenting mobile devices via outsourcing (Vallina-Rodriguez and Crowcroft, 2013; Abolfazli et al., 2013). Moreover, optimizing mobile battery lifetime requires prior knowledge on energy consuming components in mobile device, energy requirements of running mobile applications, charging and de-charging cycle of the battery, and estimation of next battery recharge. One of the most credible approaches to alleviate energy deficiency of mobile devices is to perform computational augmentation and offload energy-hungry components of mobile applications to a remote resources, as explained in Section 5.1. Mobile computation augmentation not only aims to deliver intensive computation to mobile users, but also save scarce energy and prolongs battery life. In order to realize the latter vision, researchers investigate various dimensions of energy dissipation in mobile augmentation. High communication between mobile device and remote resources, pre-offloading transactions in mobile device (e.g., resource profiling, identifying and partitioning resource-intensive codes, identifying appropriate remote resource), and overheads associated with long distance between mobile and remote devices are major sources of depleting mobile device's battery. Other transactions like monitoring, synchronizing, and integrating remote computations with native codes are part of post-offloading tasks that drains battery.

Researchers in μCloud (March et al., 2011) aim to minimize the pre-offloading overheads by executing battery-hungry components of RMAs inside the cloud. MOMCC (Abolfazli et al., 2012b) and SAMI (Sanaei et al., 2012a) exploit service oriented architecture to reduce the code partitioning and migrating overheads. In Cloudlet, Hyrax

(Marinelli, 2009), and MOMCC researchers endeavor to reduce the long WAN latency and energy cost of augmenting computing power of mobile devices using nearby computing entities. While Cloudlet uses proximate immobile computing entities such as coffee shop and Airport computers, MOMCC and Hyrax build a cloud of nearby smartphones which are able to perform computations on behalf of neighboring resource-poor smartphones. Although researchers have significantly studied varied energy efficiency approaches from different perspectives (Bianzino et al., 2012; Vallina-Rodriguez and Crowcroft, 2013; Abolfazli et al., 2013) mobile battery conservation is still one of the most crucial research agendas in mobile computing.

5.3. Storage limitation

Storage medium in majority of contemporary mobile devices is dominated by flash-based memory modules such as eMMCs (Embedded MultiMedica Cards) and micro SD cards to provide space for OS, application, and data (Kim and Kim, 2012). Low power cost, small form factor, and shock resistance are some appealing features of such memories. However, higher utilization cost compared with hard disk drives besides lower write performance compared with their read performance impact on system usability and efficiency leading to user experience degradation and RMAs failure. Thus, limitations in storage system of mobile devices are capacity (due to ever-increasing digital contents, Gantz et al., 2008) and low I/O performance (since majority of mobile online and web applications are write dominant) (Kim and Kim, 2012). While desktop computers are equipped with vast storage to store voluminous data inside local hard disks, smartphones storage is bounded to few gigabytes mostly utilized by OS, user applications, and personal content. Therefore, content management is intensified and offline usability is hindered in smartphones. To expand smartphones' local storage, researchers and industrialists employ rich cloud storages to enable unrestricted code and data access in mobile devices. For instance, Amazon S3 and Dropbox are general purpose cloud storages products offering simple operations to store and retrieve infinite data in cloud while enabling data synchronization in multiple devices.

In I/O performance improvements researchers have proposed several solutions. We review some of the academic storage solutions that expand capacity using cloud resources, and efforts that aim to enhance I/O performance of mobile devices as follows.

Phoenix (Panta et al., 2013) provides a transient distributed data storage system for mobile devices that leverages opportunistic use of cloud of nearby mobile computing devices in ad hoc manner. The authors assume existence of a cloud of mobile devices in a geographical region desireful to share their storage with other nearby mobile devices. In order to realize data consistency in mobile ad hoc environment, the authors propose an algorithm that can ensure certain degree of data redundancy over the mobile devices that participate in regional ad hoc storage system. Each data in Phoenix is broken into several blocks and each block is replicated on at least K nearby mobile nodes (e.g., at least 2 nodes) to ensure that data blocks are always available even when nodes are leaving the regional cloud or getting disconnected from the network. However, Phoenix robustness and reliability highly depend on the precise selection of K value so that data consistency can be ensured with minimum data redundancy. In this approach, advertising model is used to enable autonomous ad hoc system management and maintenance, in the absence of central administration entity. Thus, there are large redundant unicast and multicast communications between participating nodes to manage the system, which increase system complexity and networking overhead. Moreover, security, privacy, and data safety (of both mobile storage consumer and provider) in unauthenticated wireless environment originate noticeable implications that hinder usability of such system in practice. Incentivizing mobile users to share their scarce storage with other resource-constraint mobile nodes is another issue not studied in this research. Phoenix is currently working in one-hop networks only and future efforts are essential to ensure that multi-hop communication between nodes is possible.

Smartbox (Zheng et al., 2010) is a self-management, online, cloud storage and access management system developed for mobile pervasive devices to expand device storage and facilitate data access and sharing. In Smartbox, each mobile device is associated with a shadow storage to store/retrieve personal data using a unique account. To facilitate data sharing among larger group of end-users in office or at home, a public storage space is provisioned. Smartbox enhances the interaction quality by exploiting traditional hierarchical namespace for smooth navigation and employ an attribute-based method to facilitate data navigation and service query using semantic metadata (e.g., the publisher–provider metadata). Data navigation and query using tiny keyboard and small screen irk mobile users when inquiring and navigating stored data in cloud. However, mobile users need always-on connectivity to access online cloud data which is not yet achieved and demand great deal of research to become a reality.

Researchers in WhereStore (Studedi et al., 2010) propose a location-based data storage for cloud-connected smartphones to minimize the data fetch time and energy costs by replicating important cloud-stored data on the mobile device. This proposal aims to enhance context awareness, application interactivity, and energy efficiency while minimizing long WAN latency for RMAs. The fundamental concept of the WhereStore is that users entertaining in dissimilar activities in different locations require heterogeneous types of data. For example, an international visitor in Kuala Lumpur demands information related to nearby places of interest than entire Malaysia. Therefore, identifying user location besides predicting and caching essential information in timely manner could improve the system efficiency and enrich the user experience. Nevertheless, accurate prediction of future user location and his/her demanding data, and determining the right time for caching data are challenging tasks that inhibit usability and adoption of WhereStore.

Effort in Wukong (Mao et al., 2011) are to propose a cloud-based user-friendly, highly available file service for multitude of heterogeneous mobile devices. It realizes the vision of opening and modifying cloud-stored contents from mobile application without downloading them into the native mobile device storage. Utilizing a service abstraction layer (SAL), the researchers provision support of heterogeneous cloud-side services such as FTP and Google docs service in a transparent manner and propose the cache management and pre-fetch mechanisms in varied scenarios to enhance system performance and reduce long WAN latency. Nevertheless, it fails to always minimize long WAN latency because of limited wireless bandwidth and costly I/O operations. Pre-fetching data from cloud to the smartphone can reduce WAN latency and save energy toward rich user experience when there is a long gap between open and read operations. Moreover, Wukong alleviates data security and shrinks data communication volume by employing an encryption and compression mechanisms. Although compression methods leveraged in this work is beneficial for text and log files compression, it is less effective when compressing multimedia files such as image, music, and video.

ADVIL (A Device driVer for Improving Lame storage performance) (Kim and Kim, 2012) is a software-layer for OS kernel aims to mitigate the overhead of frequent random writes in the physical storage without modifying the existing file systems. Advil as a block-level device driver works as a middleware between mobile file system and flash-based storage medium to transform random write requests to sequential write requests. For each partition called *original area* on the medium, Advil builds a high speed buffer area called *reserved area* that can log filter small write instructions in a transparent manner to the mobile file system and flash-based storage devices. Advil logically partitions the reserved area into log and hot regions. Upon receiving of write instruction, Advil write them sequentially to the log region instead of directly writing in the original area. When the log area is consumed, its hot contents (data that will be overwritten in the future) are appended to the hot region and cold data are written to the original area. Such mediation role improves the overall write performance of mobile storages up to 3 times.

5.4. Visualization deficiencies

Developing rich UI and interaction experience in RMAs is a non-trivial task due to small visualization area and rendering limitations of mobile devices. Small screen size obstructs data visualization, especially large data, while rendering constraints prohibit utilization of resource-intensive, high-end data presentation techniques. To mitigate these problems, academic and industrial communities endeavor to augment smartphones' visualization abilities. We classify them into two major groups of visualization area extension and content adaptation that are described as follows.

5.4.1. Visualization area extension

Kyocera Communications manufactured a double 3.5-inch screen smartphone named Echo[1] envisioning to extend shared screen size of smartphones to 4.7 inches diagonally. Such device is built to be used in different functioning modes so that screens can be used either individually or together. However, adoption of such devices is hindered due to their form factor and energy dissipation issues that demand future R & D efforts. GeoTV (Yih-Farn et al., 2009) and the Nirvana phone (Open Kernel Labs and Citrix Systems, 2009) utilize proximate, large screens similar to a TV or computer monitor to expand smartphone presentation area. Yet, these solutions require immobile infrastructure like a TV that would immobilize the smartphones. Alternatively, ad hoc solutions such as the mobile projector used in Samsung Galaxy Beam[2] improve smartphone data visualization and enable mobile users to beam and share photos, videos, and presentation slides on a 50-inch wide area wall and such. Although the battery lasts for about 3 h and the projection quality is not as fine as a modern projector's — especially in a bright place — it can be beneficial in scenarios where people desire to share their screen. Clearly, screen projection is not a suitable solution when looking at private, sensitive information.

Mobile 3D display technology can overcome visualization problems if computing and power limitations of smartphones are mitigated. Glass-free auto-stereoscopic displays (Ortiz, 2011) are designed to present 3D data via binocular parallax that offers a different view for each eye. However, this is computationally more complex than 2D display (Capin et al., 2008). Moreover, several soft techniques, including tabular pages, 3D objects, multiple desktops, switching between landscape and portrait views (needs accelerometer sensor), and verbal communication can increase presentation area and enrich quality of interaction in contemporary smartphones.

Virtualized screen (Lu et al., 2011) aims to enhance UI rendering capabilities by moving screen rendering processes to the cloud and produce the rendered screen as an image to the mobile device. The authors assume that most of the compute- and data-intensive computations take place in the cloud and so it is feasible to enrich the user experience and migrate the screen rendering tasks to the cloud. Therefore, exploitation of infinite clouds can augment computing capabilities of smartphones, prolongs device battery, and enhances the interaction and responsiveness of mobile applications toward rich user experience (Abolfazli et al., 2013). Screen virtualization technique (running partial UI rendering in cloud and remaining in the mobile device) can enrich user experience, especially for lightweight, high-fidelity, and interactive mobile applications that can be executed on the local resources. The authors in this proposal aim to augment visualization abilities of mobile devices, reduce the impacts of heterogeneity among multitude of mobile devices, and ease porting RMAs to dissimilar mobile devices (e.g., smartphone, laptop, and IP TV) with different screen sizes (Sanaei et al., 2013). To mitigate the overhead of mobile-cloud data transmission, a frame-based delegate system is maintained to forward the visual updates from the cloud to the smartphones. Frame-based delegate system captures and feeds the entire screen image to the transmission unit. In this work, frames are updated according to the changes in previous frame stored inside the mobile and cloud device. However, interactivity and responsiveness requirements of RMAs obligate online streaming of screen images which is impacted by long WAN latency. Although optimized screen transmission approaches are deployed to reduce the networking traffic, the impacts of computation and communication latencies are not yet investigated in this preliminary solution.

5.4.2. Content adaptation

Mobile and immobile applications feature dissimilar scope, aims, and contents abstraction levels. Comprehensive contents are desirable in immobile machines, whereas compact contents are appropriate for smartphones (Murugesan and Deshpande, 2002). However, it is not feasible to produce different content versions for various devices and accurate automatic content adaptation is associated with two challenges of semantic gap and user intention gap. To realize accurate content adaptation for multitude of heterogeneous mobile devices, researchers in Yin et al. (2011) leverage event-based semantic image adaptation scheme to eliminate unnecessary or less important contents based on user preference and available visualization areas. In this approach, an event-dependent semantic extraction method aims to identify important people or objects and associate them with each other in any given image. Also, a simple feedback mechanism dynamically collects user updated preference context to integrate them with the information provided during semantic extraction. Accordingly, an adaptation algorithm integrates important objects with user preferences to produce a suitable version for target mobile device.

Chen et al. (2005) propose and implement a page split method inside of the mobile browser to adapt the web page contents to the mobile devices according to the screen size and content semantics. The web page can be further processed to build a two-level hierarchy with thumbnail in top level and global view at bottom level. During a 3-step page analysis mechanism, they exploit production-time layout of web pages to split the page into small logically related blocks. In the first step HTML tags are analyzed to specify high-level content blocks. In second step, contents in each high-level contents block are analyzed to identify explicit page separators and in last step, implicit separators are analyzed to accurately split the contents in a particular page. Although results are promising, components like page analysis, content detection, page splitting, and index-page generation algorithms consume high volumes of local resources.

5.5. Data safety and security threats

Establishing trust in RMAs becomes extensively complex due to the insecure nature of smartphones in wireless mobile

[1] http://www.echobykyocera.com/
[2] http://www.samsung.com/in/galaxybeam/feature.html

environment. Mobile devices are susceptible to safety breaches because of high probability of hardware malfunction, physical damage, stealing, and loss. Hence, information stored inside the local storage of mobile devices is not safe. Moreover, common mobile malwares such as virus, Trojans, rootkits, and botnets (La Polla et al., 2012) besides future threats, specifically financial malware and surveillance attacks deteriorate security and privacy of mobile users, especially Android users.

Additionally, the need for simple UI in RMAs and neglecting security principles by end-users increase security risks of mobile computing (Srivastava and Nandi, 2013). For instance, hiding a bank's web link beneath an icon helps the mobile user to read and login quickly, but a hidden bank address is susceptible to being changed to a fake link without the user perceiving it (Whitten, 2004; Bratus et al., 2008). Moreover, features like GPS and accelerometer in smartphones potentially can violate user security and privacy (Marquardt et al., 2011). For example, authors in (SP) iphone demonstrate that utilizing smartphone's accelerometer sensor, the malicious code can recover text entered on a nearby keyboard. They could extract as high as 80% of the text by detecting and decoding keystrokes while measuring the physical position and distance between different vibrations.

Researchers in Enck et al. (2011) aim to augment trust among mobile users by analyzing more than 300 Android applications to identify attacking points and approaches in mobile application. The authors develop an application certification model to automatically distinguish malware from safe applications. However, the usability of this algorithm cannot be extended much due to the presence of complicated malware that cannot be spotted easily. Moreover, mobile devices are not a safe medium for storing sensitive user data (e.g., banking and financial data) due to their security vulnerabilities. To enhance data safety, researchers employ remote storage to store data on remote servers which are likely safer than mobile devices. However, cloud resources are vulnerable too. Several efforts (Kamara and Lauter, 2010; Wang et al., 2009) by various researchers were meant to investigate cloud storage security and propose methods to better secure cloud storage; however, breaches like Amazon EC2 crash and Dropbox security glitch are still reported (Cachin and Schunter, 2011).

Hence, ensuring data safety, security, and privacy of mobile devices in wireless environment has become extensively complex. This insecure platform makes smartphones a less trustworthy ground and hence, slows down the adoption and development of RMAs.

5.6. Intuitive interaction methods

Providing rich interaction in RMAs is a challenging task in the absence of intuitive interaction devices. Without mouse, intuitive methods such as selection, double click, and right click are redefined or even undefined in smartphones; if the object is touched, execution is initiated. Similarly, double clicking is dismissed, the dragging gesture is transformed to scrolling and right clicking is virtually simulated by a long touch in several OSs, particularly iOS and Android.

The keyboard is a common text-based data entry method in the majority of smartphones, but data entry using a tiny keyboard is an irksome impediment to users. However, several features such as handwriting recognition, camera, data collecting sensors, GPS, and accelerometer can be leveraged for collecting data that noticeably enhance the quality of interaction in modern smartphones (Ballagas et al., 2006). For instance, Samsung maintains 3-axis accelerometer and 3-axis magnetic sensor to provide 3D motion-detection and enable device to collect data when user is moving, a computing task that is resource-intensive. We present some of the efforts that aim to enrich interaction quality of mobile users as follows.

Haro et al. (2005) aim to facilitate mobile-user interaction in the absence of mouse, using the smartphone camera. A mobile-user can interact with an application by moving his/her device while the camera captures user reflection and movement, a feature-based tracking algorithm analyzes them and determines the scrolling directions and magnitudes. The collected data are sent to the event handler — identical to the same component used in detecting mouse movement— to determine the scroll direction and level. The camera is leveraged as a pointing device and zoom level control in applications, similar to mouse function. Hence, it can meet the requirements of mouse-driven applications and mouse-oriented users. However, accurate estimation of camera movements in dynamic, rapidly-changing, mobile environment is challenging. For instance, if user moves in front of camera without moving the device, the system is unable to identify the source of movement and fail to accurately identify the scrolling direction and level.

WalkSafe (Wang et al., 2012) is an Android application that aims to increase the interaction quality of mobile users with their external environment. It utilizes smartphone's back camera and accelerometer sensor to enhance the walking safety of mobile talking pedestrians. While the user is talking on the phone, the camera continuously captures and analyzes captured images to alert users if a car is approaching. However, WalkSafe functionality heavily relies on the vision capabilities of smartphones. So, users are at risk if they obscure the camera vision by finger, scarf, or any other object while talking. Likewise, if an approaching car is not in the camera's view or the surrounding light is not sufficient to capture a good photo, user safety is in danger. These methods are also not energy-efficient; in case of WalkSafe, the application turns the camera on while the user is talking and its decision making is based on an image recognition process including several steps of capturing, pre-processing, and analyzing captured images via an offline dataset which rapidly drains the battery. Leveraging cloud-based augmentation models (Abolfazli et al., 2013) in MCC can significantly improve the interaction quality and reliability in such systems.

NeuralPhone is a brain-to-smartphone interface (Campbell and Choudhury, 2012) that enables intuitive interaction with smartphone applications without touching the device. They leverage a non-expensive digital device (called electroencephalography) to process brain's signals and transform them into wireless signals understandable by the smartphone. The authors have implemented an address book application which can flash the photos from the contact lists, when the photo matches the user decision, the phone automatically dial the number. However, it requires user to wear a bulky electroencephalography device to read the neural signals and look at the images to pick a particular contact. Users in future NeuralPhone may think about a friend and smartphone can dial the number without browsing all the contact list's photos.

5.7. Heterogeneity

Delivering rich functionality, crisp response, data ubiquity, and offline usability in heterogeneous mobile computing environment are non-trivial tasks in the absence of standards, technologies, and heterogeneity handling solutions (Sanaei et al., 2013). Heterogeneity in mobile computing arises due to the existence of various non-similar and non-uniform mobile devices, developing platforms, and networking technologies. Smartphone vendors develop various approaches and technologies to enhance the quality of their products, which significantly complicates porting a RMA to different computing devices. Existing mobile OSs such as Android, iOS, and Symbian are developed with different operating architectures, programming languages, and API levels. Even a single OS, in particular Android comes in various versions with different API levels. Therefore, mobile application developers should develop multi-versions of a single mobile application, each for a particular mobile device — which is a non-trivial task. Developing one application for several platforms is a costly process and demands extensive knowledge of several OSs and programming languages, and enforces redundant design, programming, testing, and evaluating tasks.

Microsoft, Adobe, and Oracle have been proposing heterogeneous developing technologies and platforms to deliver richness to end-users. Each of these technologies focuses on certain richness aspects. For instance, Adobe Flex enables developers to build cross-platform online applications. Java FX Mobile creates feature-rich applications with emphasis on security and portability. Using Java FX mobile, one can run an application on desktop, mobile, and any other device regardless of screen size, shape, and quality. However single mobile OS does not support all available tools and technologies; similarly there is no single developing platform usable for all mobile devices. Heterogeneity problems are intensified in RMAs due to their dependency to remote computational resources, especially heterogeneous cloud-based resources. Heterogeneity-made problems such as portability and interoperability among cloud-based resources complicate development of RMAs. Therefore, further research and study are required to facilitate portability and interoperability between heterogeneously built RMAs. Some of the efforts and proposed solutions to alleviate heterogeneity toward enhancing RMAs development are described as below.

Mirage (Madhavapeddy et al., 2010) is an OS kernel (exokernel, Engler, 1998) that enables developers to produce rich, multi-tier applications on normal OSs like Linux and port them to large number of mobile phones and cloud resources. Mirage can automatically convert the application to a fully standalone cross-platform application. The converted application runs on large variety of devices like desktop, smartphone and TV regardless of underlying platform's heterogeneity. Mirage compiles and binds mobile application into the kernel installed and directly runs on ARM-based mobile devices and Xen-based clouds.

Researchers in Loutas et al. (2010) perceive interoperability issues between cloud systems and argue that convergence of SOA and semantics can herald a new interoperable cloud landscape. The authors describe RASIC which is an architecture for heterogeneous clouds that realizes semantic interoperation among inhomogeneous clouds. A user-centric paradigm is proposed to enhance development and deployment of SOA-based services in a large-scale, resource-intensive environment hosted by different cloud providers. Utilizing SOA design philosophy facilitates content movement across heterogeneous clouds and reduces the cost of data and code porting from one cloud to another.

Researchers (Ranabahu and Sheth, 2010) propose a solution for portability and interoperability by employing semantics in the cloud resource to overcome three fundamental problems of defining functionalities, modeling data, and facilitating service description. The authors propose a semantic-centric programming paradigm by identifying four types of semantics for an application, namely system, non-functional, data, and logic & process. However, utilizing semantics in MCC is not straightforward due to intensive processing and reasoning tasks besides challenges of acquiring and integrating large, reliable, and accurate semantic data (Sabou, 2010; Corcho and García-Castro, 2010).

Moreover, several cross-platform solutions like Marmalade and PhoneGap automatically transit applications to other platforms by code regeneration. The mobile application is developed once and transited to many devices with different OSs. However, according to American National Standard Dictionary of IT, portable applications should be able to migrate among different platforms with *no/less modification and configuration* which is yet an open research area.

5.8. Communication and mobility limitations

Communication latency and seamless mobility in mobile environment directly impacts on the RMAs' richness degree from different aspects. Communication latency in wireless medium is directly affected by three main factors of processing, transmission, and propagation delay. Such accumulated delay prolongs the application execution and data traveling time between sender and receiver and wastes more resources due to excessive monitoring efforts and redundant tasks like compressing, encrypting, and marshaling (Whaiduzzaman et al., 2013). Latency is exacerbated by client's mobility during the communication due to change in distance, number of intermediary nodes, bandwidth, and heterogeneous wireless technologies.

Moreover, wireless bandwidth is very limited compared to the fixed lines that challenges bulky data transfer between mobile device and remote server. Also, each technology in silo of heterogeneous wireless technologies provides different bandwidth. Such inconstant bandwidth likely causes bottleneck in several intermediary points that leads to excessive packet drop, latency, and battery drainage. Therefore, transmitting data and code to/from mobile devices takes excessive time which wastes energy and degrade interactivity and responsiveness of RMAs.

Nevertheless, realizing seamless mobility in mobile environments to provide pervasive, convenient, on-demand wireless access to cloud-based computing resources is non-trivial challenge. Seamless code mobility intensifies because both service providers and consumers can move during the application execution time, which intensifies code mobility (Yu et al., 2012). Consequently, networking interruption and intermittency cause several issues, especially lack of always-on connectivity, excessive native mobile resource consumption, and frequent application interruption and failure which lowers quality of computing services and degrades RUX (Satyanarayanan, 2005). Also, it originates redundant costs on cloud-connected mobile users and encumbers reliability. Therefore, alleviating mobility using Web advancements (Johansson and Andersson, 2012) and emerging lightweight cognitive mobility management systems with minimum signal traffic and latency can noticeably improve ubiquitous connectivity.

Among efforts to alleviate communication difficulties, Cloudlet (Satyanarayanan et al., 2009) intends to achieve satisfaction and offer high quality, crisp response to delay-sensitive mobile applications by performing remote computing-intensive tasks. MOMCC is a recent effort employing crowd of proximate smartphones to minimize the impact of transmission latency between mobile client and server(s). MOMCC is a market-oriented, SOA-based architecture that enables service-based applications to be executed on smartphones. The fine-grained services are developed by expert programmers and hosted by layman users on smartphone. In runtime, the service end-user/requester demands execution of the required service on the nearest host to tackle the transmission latency. SAMI provides a multi-layer architecture for MCC that allocates appropriate type of resources to each task using a resource scheduler component. For instance, a delay-sensitive service will be hosted and executed on MNO dealer's machine, while resource-intensive service runs inside the cloud. However, MOMCC and SAMI are preliminary works and several challenges need to be addressed.

6. Open issues

This section presents several important issues that have not been thoroughly studied till now as research directions in the development and adoption of RMAs.

1. *Architectural*: Design methodology for appropriate selection of layers distribution in RMAs is a significant thought-provoking open issue. RMAs are exposed to a wide range of domains, particularly data processing, office automation, enterprise applications, healthcare, and education — each of which with certain requirements. For instance, enterprise applications require rich functionality and offline usability while healthcare applications demand responsiveness and online connectivity. Therefore, accurate decision making on where/when/how to distribute the user interface, logic, and database layers and need for generic reference architecture to address architectural requirements of a wide array of RMAs in future is essential.
2. *Platform neutrality*: Producing platform-neutral RMAs is hindered by hardware, features, OSs, programming languages, and network heterogeneity. The tight dependency of RMAs to the underlying platforms stems several problems, mainly code and data fragmentation, vendor lock-in, and portability problems. In turn, these problems divest the "write once run anywhere" principle from the developer and lock user data and applications in specific device(s) or server(s). Hence, a RMA written for a specific device is unable to be migrated to different platform (s) and device(s) with no/minor modification, though several cross-platform development and transition solutions have been proposed. Leveraging service oriented cloud-based infrastructure similar to Sanaei et al. (2012a) and Abolfazli et al. (2012b) deemed to be beneficial in developing loosely coupled and platform-independent RMAs, which is granting new research area.
3. *Resource scheduling*: Rich functionality of RMAs requires adequate allocation of resources from the pool of heterogeneous cloud-based computational resources to perform heavy computations on behalf of the resource-limited smartphones with least overhead. Varied types and multiplicity of cloud-based resources obligate a robust resource scheduling algorithm that can optimally allocate computing resources to the computational tasks based on multiple objectives and criteria. Communication latency, computation performance and capability, reliability, security, and utilization cost are major resource scheduling criteria to be considered for future resource scheduling solutions. Multi-objective algorithms such as Rahimi et al. (2012) and Delimitrou and Kozyrakis (2013) are inevitable need of future RMAs.
4. *Context awareness*: Although context- and social-aware RMAs in sensor-rich, contemporary smartphones deliver RUX to mobile users, processing, managing, and interpreting voluminous acquired context information by resource-constrained smartphones are challenging tasks. Despite the extensive study of context awareness (Hong et al., 2009), solutions that fully realize this in a rapidly changing mobile environment are insufficient (Truong and Dustdar, 2009). Resource-efficient, smartphone-friendly approaches that exploit context information with minimum footprint are essential for rich mobile computing to flourish. Collecting highly dynamic context information such as location, humidity, weather, and user preferences in the dynamic mobile landscape is not difficult (Conti et al., 2012), but such significant volumes of information are arduous to manage, process, and interpret (Conan et al., 2007; Truong and Dustdar, 2009). Additionally, with regard to user mobility, integrating changes with current content, UI, and services breeds extra challenging tasks. Therefore, context awareness is still an open research domain in smartphones, though plenty of efforts exist on context-aware desktop applications (Zhang et al., 2011b; Bellavista et al., 2013).
5. *Seamless ubiquity*: Non-continuity and non-consistency are two important weaknesses of wireless communication that hinder seamless connectivity and deprive nomadic users from rich ubiquity due to network intermittency. Irregular connectivity causes several challenges such as dismissal of always-on connectivity, excessive consumption of limited mobile resources, and disproportionate delay of application execution that sharply degrade quality of mobile computing services and decrease RMAs' interaction response quality. Several issues in heterogeneous wireless technologies (e.g., WLAN, 2G, and 3G), particularly signal handoff, jitter, and long WAN latency slow down RUX delivery to mobile-cloud users. An inefficient, blipfull wireless medium prolongs application execution time, and consequently drains local resources, especially the battery. Forthcoming technologies, particularly cognitive radio (Thomas et al., 2005), Open Wireless Architecture (OWA) (Lu et al., 2012), and context-aware communication (Proebster et al., 2011) are expected to provide seamless and efficient mobility (device, data, and environment) in heterogeneous communication environments to enhance QoS and QoE.
6. *Intensive computation*: Delivering rich functionality to RMAs' users is a non-trivial task due to resource deficiency of smartphones. Although several efforts have been made to augment mobile devices, complex functionality of RMAs is not fully realized. Smartphone CPUs have become significantly more powerful, but memory, storage, and energy in particular are still noticeably limited. In some scenarios, smartphone computing and networking capabilities can fulfill RMAs' resource requirements, but limited energy hinders such capability. Smartphone power management and awareness remain subjects of further research and development (Cui et al., 2012).

   While delivering rich functionality in collaborative computing environment via crowd of nearby wireless computing devices can enrich user experience (Abolfazli et al., 2012b), several open issues with energy, computation, and communications subsystems such as performance evaluation, efficient message broadcasting, routing, and synchronization need to be addressed (Datla et al., 2012). Hence, lightweight technologies and mobile architectures similar to MOMCC, and SAMI are needed to create intensive applications without considering current smartphone flaws.
7. *Mobile threats*: Security and privacy have been challenging research areas, especially in the wireless environment (Omar et al., 2012; von Mulert et al., 2012). Rapid proliferation of smartphones and pervasiveness of open-source mobile OSs such as Android intensify mobile security threats. The attacking vectors in contemporary mobile devices and malware structures have totally changed due to the existence of several new features, mainly camera, GPS, and accelerometer. Kaspersky[3] predicts significant growth in mobile malware and attacks in coming years, mostly targeting Google mobile OS. User privacy is also vulnerable despite the presence of cloud storage such as Dropbox and Amazon S3. Hence, securing mobile devices, protecting user privacy, and increasing data safety as new challenging research areas — that affect RMA usability — require vast research and development in coming years. Lightweight security approaches by utilizing outsourcing MCC solutions and novel security methods using techniques other than traditional cryptography (such as data partitioning, Leistikow and Tavangarian, 2013) can realize security in mobile computing.
8. *User-centric recommendation*: RMAs are potentially suitable for offering user-centric computing to mobile users. Mobile users' personal preferences, social traits, and competence impact QoE while interacting with mobile applications. However, exponential growth in the number of mobile applications tampers the selection of suitable applications for particular requirement. Thus, user-centric application recommender systems like Woerndl et al. (2007) are vitally needed. Providing a proper scale to measure

[3] http://www.kaspersky.com

the degree of richness can be beneficial to adequately assigning a grade to each RMA based on application characteristics (e.g., energy efficiency, responsiveness, interactivity, offline usability, and context awareness), user preferences, and user feedback. These grades will be utilized by the recommender systems to advise prospectus users regarding the most appropriate applications.

7. Conclusions

Significant achievements in cornerstone technologies, specially wireless communication, mobile computing, and cloud computing on one hand and momentous advances in contemporary mobile devices, particularly smartphones on the other hand pave the way toward realizing vision of rich computing anywhere at anytime. Traditional mobile application models cannot fulfill such insatiable computing requirements of mobile users, which encourages academician and practitioners to enrich mobile applications by employing contemporary technologies of mobile and distributed computing domains, particularly cloud computing that breeds Rich Mobile Applications (RMAs). RMAs are deemed could be the future model of mobile applications which are featured with energy efficiency, high functionality, rich interactivity, crisp responsiveness, and seamless ubiquity — to name a few. RMAs comprise suitable ground for a plethora of domains, particularly enterprise and business, information management, healthcare, emergency handling, and crowd management.

In this paper, RMAs were comprehensively surveyed along with recent developments in delivering RUX to mobile users. RMAs' genesis is analyzed and illustrated, and a definition is provided. Richness characteristics of RIAs — as the predecessor— were synthesized with RMAs. Interactivity and context awareness are two RMAs features that are distinct from those of RIAs. RIAs deliver a higher degree of functionality, security, offline usability, and online connectivity in a fixed place to end-users, whereas RMAs offer high functionality, seamless ubiquity, context-awareness, and immersive interaction on the go. While RMAs' energy efficiency vision to enhance the quality of user experience by prolonging battery lifetime, that of RIAs is aimed for green and economical computing via reduced power consumption. Hence, richness in RMAs is a unique property, something that is not achievable by traditional mobile applications. Richness in mobile applications requires extensive processing capabilities, immersive developing platforms, and seamless communication. However, several problems such as processing limitations of mobile devices, battery constraint, storage limitations, heterogeneity, security, and seamless mobility inhibit the usability of smartphones as a reliable platform for RMAs. These problems were analyzed and classified into eight classes and available solutions were surveyed to devise a taxonomy. Despite advances in various dimensions, several open issues, especially RMAs architecture, context-awareness, seamless ubiquity, and intensive computation need to be investigated and addressed in future.

Acknowledgment

This work is funded by the Malaysian Ministry of Higher Education under the University of Malaya High Impact Research Grant - UM.C/HIR/MOHE/FCSIT/03.

References

Abolfazli S, Sanaei Z, Gani A. Mobile cloud computing: a review on smartphone augmentation approaches. In: Proceedings of WSEAS CISCO'12, Singapore, 2012.

Abolfazli S, Sanaei Z, Shiraz M, Gani A. MOMCC: Market-oriented architecture for mobile cloud computing based on service oriented architecture. In: Proceedings of IEEE MobiCC'12, Beijing, China, 2012. p. 8–13.

Abolfazli S, Sanaei Z, Ahmed E, Gani A, Rajkuma B. Cloud-based augmentation for mobile devices: motivation, taxonomies, and open issues, IEEE Communications Surveys & Tutorials, June 2013, http://dx.doi.org/10.1109/SURV.2013.070813.00285.

Ahlgren B, Dannewitz C, Imbrenda C, Kutscher D, Ohlman B. A survey of information-centric networking. IEEE Communications Magazine 2012;50(7):26–36.

Akyildiz IF, Jiang X, Mohanty S. A survey of mobility management in next-generation all-IP-based wireless systems. IEEE Wireless Communications 2004;11(4):16–28.

Almenares F, Arias P, Marin A, Diaz-Sanchez D, Sanchez R. Overhead of using secure wireless communications in mobile computing. IEEE Transactions on Consumer Electronics 2013;59(2).

Andruszkiewicz P, Rybiski H, Protaziuk G, Gajda M. How rich are mobile rich internet applications? Emerging Intelligent Technologies in Industry 2011;369:283–95.

Anegg H, Dangl T, Jank M, Niklfeld G, Pucher M, Schatz R, et al. Florian Wegscheider, Multimodal interfaces in mobile devices—the MONA project. In: Proceedings of MobEA'04, New York, USA, 2004.

Ballagas R, Borchers J, Rohs M, Sheridan JG. The smart phone: a ubiquitous input device. IEEE Pervasive Computing 2006;5(1):70–7.

BCC Research. Global markets for smartphones and PDAs. Technical Report; 2009.

Bellavista P, Coeeadi A, Fanelli M, Foschini L. A survey of context data distribution for mobile ubiquitous systems. ACM Computing Surveys 2013;45(1).

Ben D, Ma B, Liu L, Xia Z, Zhang W, Liu F. Unusual burns with combined injuries caused by mobile phone explosion: watch out for the "mini-bomb!". Journal of Burn Care and Research 2009;30(6):1048.

Bianzino AP, Chaudet C, Rossi D, Rougier J-L. A survey of green networking research. IEEE Communications Surveys & Tutorials 2012;14(1):3–20.

Birman KP. Building secure and reliable network applications. Manning Publications Co.; 1997.

Bloch J. How to design a good API and why it matters. In: Proceedings of ACM OOPSLA'06, 2006. p. 506–7.

Bondi AB. Characteristics of scalability and their impact on performance. In: Proceedings of ACM WOSP'00, New York, USA, 2000. p. 195–203.

Bratus S, Masone C, Smith SW. Why do street-smart people do stupid things online? IEEE Security & Privacy 2008;6(3):71–4.

Broberg J, Buyya R, Tari Z. MetaCDN: harnessing storage clouds for high performance content delivery. Journal of Network and Computer Applications 2009;32(5):1012–22.

Burrell G, Morgan G. Sociological paradigms and organizational analysis, vol. 12, 1979.

Cachin C, Schunter M. A cloud you can trust. IEEE Spectrum 2011;48(12):28–51.

Campbell A, Choudhury T. From smart to cognitive phones. IEEE Pervasive Computing 2012;11(3):7–11.

Campesato O, Nilson K. Web 2.0 fundamentals with Ajax, development tools, and mobile platforms. Jones and Bartlett Publishers, 2011.

Capin T, Pulli K, Akenine-Moller T. The state of the art in mobile graphics research. IEEE Computer Graphics and Applications 2008;28(4):74–84.

Chen Y, Xie X, Ma WY, Zhang HJ. Adapting web pages for small-screen devices. IEEE Internet Computing 2005;9(1):50–6.

Cho S-J, Choi C, Sung Y, Lee K, Kim Y-B, Murray-Smith R. Dynamics of tilt-based browsing on mobile devices. In: Proceedings of ACM CHI'07 extended abstracts on human factors in computing systems, San Jose, CA, USA, 2007. p. 1947–52.

Christodoulou SP, Styliaras GD, Papatheodorou TS. Evaluation of hypermedia application development and management systems. In: Proceedings of ACM hypertext'98, Pittsburgh, PA, USA, 1998. p. 1–10.

Chun B-G, Maniatis P. Dynamically partitioning applications between weak devices and clouds. In: Proceedings of ACM MCS'10: social networks and beyond, San Francisco, USA, 2010. p. 1–5.

Chun BG, Ihm S, Maniatis P, Naik M, Patti A. CloneCloud: elastic execution between mobile device and cloud. In: Proceedings of ACM EuroSys'11, Salzburg, Austria, 2011. p. 301–14.

Conan D, Rouvoy R, Seinturier L. Scalable processing of context information with COSMOS. In: Proceedings of springer-verlag DAIS'07, 2007. p. 210–24.

Conti M, Das SK, Bisdikian C, Kumar M, Ni LM, Passarella A, et al. Looking ahead in pervasive computing: challenges and opportunities in the era of cyber-physical convergence. Pervasive and Mobile Computing 2012;8(1):2–21.

Corcho O, García-Castro R. Five challenges for the semantic sensor web. Semantic Web 2010;1(1):121–5.

Cormode G, Krishnamurthy B. Key differences between Web 1.0 and Web 2.0. First Monday 2008;13(6):1–30.

Cuervo E, Balasubramanian A, Cho D, Wolman A, Saroiu S, Chandra R, et al. MAUI: making smartphones last longer with code offload. In: Proceedings of ACM MobiSys'10, San Francisco, CA, USA, 2010. p. 49–62.

Cui Y, Ma X, Wang H, Stojmenovic I, Liu J. A survey of energy efficient wireless transmission and modeling in mobile cloud computing. Mobile Networks and Applications 2012:1–8.

Datla D, Chen X, Tsou T, Raghunandan S, Hasan SMS, Reed JH, et al. Wireless distributed computing: a survey of research challenges. IEEE Communications Magazine 2012;50(1):144–52.

Deb B, Bannur SG, Bharti S. Rich internet applications (RIA): opportunities and challenges for enterprises. Technical Reports, Infosys; 2007.

Delimitrou C, Kozyrakis C. Paragon: QoS-aware scheduling for heterogeneous datacenters. In: Proceedings of ASPLOS'13, Texas, USA, 2013. p. 77–88.

Dey AK. Understanding and using context. Personal and Ubiquitous Computing 2001;5(1):4–7.

Duhl J. Rich internet applications. Technical Reports, IDC Corporate USA; 2003.

Duo L, Serizawa Y, Kiuchi M. Extension of client–server applications to the internet. In: Proceedings of TENCON, vol. 1, 2002. p. 355–358.

Emmanouilidis C, Koutsiamanis R-A, Tasidou A. Mobile guides: taxonomy of architectures, context awareness, technologies and applications. Journal of Network and Computer Applications 2013;36(1):103–25.

Enck W, Ongtang M, McDaniel P. On lightweight mobile phone application certification. In: Proceedings of IEEE CCS'11, Milan, Italy, 2011. p. 235–45.

Engler DR. The Exokernel operating system architecture [PhD thesis]. Massachusetts Institute of Technology, 1998.

Eslambolchilar P, Murray-smith R. Tilt-based automatic zooming and scaling in mobile devices—a state-space implementation. In: Proceedings of springer mobile HCI'04, vol. 3160, Glasgow, UK, 2004. p. 120–31.

Ferguson S, Beilharz K, CalÃ C. Navigation of interactive sonifications and visualisations of time-series data using multi-touch computing. Journal on Multimodal User Interfaces 2012;5(3):97–109.

Fletcher RA, MacDonald NB. Portability and code reuse in parallel applications. In: Proceedings of the international conference on parallel computing (ParCo'93), Grenoble, France, 1993.

Flinn J, Satyanarayanan M. Energy-aware adaptation for mobile applications. In: Proceedings of ACM SOSP's 99, vol. 33, 1999. p. 48–63.

Forrester Research. eCommerce Website performance today, an updated look at consumer reaction to a poor online shopping experience. eCommerce Website performance today, an updated look at consumer reaction to a poor online shopping experience, commissioned study. USA: Forrester Research, Inc; 2009.

Fraternali P, Rossi G, Sánchez-Figueroa F. Rich internet applications. IEEE Internet Computing 2010;14(3):9–12.

Gantz JF, Chute C, Manfrediz A, Minton S, Reinsel D. Schlichting W, et al. The diverse and exploding digital universe: an updated forecast of worldwide information growth through 2011, White Paper, 2008.

Gartner reveals top predictions for IT organizations and users for 2012 and beyond; December 2011. URL: ⟨http://www.gartner.com/newsroom/id/1862714⟩.

Gartner says worldwide PC, tablet and mobile phone combined shipments to reach 2.4 billion units in 2013; 2013. URL: ⟨http://www.gartner.com/newsroom/id/2408515⟩.

Gruman IG, Mobile management: InfoWorld's expert guide. URL: ⟨http://infoworld.com/d/mobilize/mobile-management-infoworlds-expert-guide-371-0⟩.

Hariharan K. Best practices: extending enterprise applications to mobile devices. The Architecture Journal Microsoft Architecture Center 2008;14.

Haro A, Mori K, Capin T, Wilkinson S. Mobile camera-based user interaction. In: Computer vision in human–computer interaction. Lecture notes in computer science, vol. 3766. Berlin, Heidelberg: Springer; 2005. p. 79–89.

Hong J, Suh E, Kim SJ. Context-aware systems: a literature review and classification. Expert Systems with Applications 2009;36(4):8509–22.

Hung S-H, Shih C-S, Shieh J-P, Lee C-P, Huang Y-H. Executing mobile applications on the cloud: framework and issues. Computers and Mathematics with Applications 2011;63(2):573–87.

Johansson D, Andersson K. Web-based adaptive application mobility. In: Proceedings of IEEE CLOUDNET'12, Paris, 2012. p. 87–94.

Jones E, Alexander J, Andreou A, Irani P, Subramanian S. GesText: accelerometer-based gestural text-entry systems. In: Proceedings of ACM CHI'10, Atlanta, GA, USA, 2010. p. 2173–82.

Kaasinen E. User needs for location-aware mobile services. Personal and Ubiquitous Computing 2003;7(1):70–9.

Kamara S, Lauter K. Cryptographic cloud storage. Financial Cryptography and Data Security 2010;20:136–49.

Kemp R, Palmer N, Kielmann T, Bal H. The smartphone and the cloud: power to the user. In: Proceedings of MobiCASE'12, California, USA, 2012. p. 342–8.

Kim HS. Location-based authentication protocol for first cognitive radio networking standard. Journal of Network and Computer Applications 2011a;34(4):1160–7.

Kim J-M, Kim J-S. Advil: a pain reliever for the storage performance of mobile devices. In: 2012 IEEE 15th international conference on computational science and engineering, IEEE, 2012. p. 429–36. http://dx.doi.org/10.1109/ICCSE.2012.66.

Kim RY, Mohanty S. Advanced power management techniques in next-generation wireless networks. IEEE Communications Magazine 2010;48(5):94–102.

King, R. Google cleans up user interface on gmail offline, mobile web app; 2013. URL: ⟨http://goo.gl/JvqPv⟩.

Koukoumidis E, Martonosi M, Peh L-S. Leveraging smartphone cameras for collaborative road advisories. IEEE Transactions on Mobile Computing 2012;11(5):707–23.

Kramer G. Auditory display: sonification, audification, and auditory interfaces. Addison-Wesley; 1994.

Lane ND, Miluzzo E, Hong L, Peebles D, Choudhury T, Campbell AT. A survey of mobile phone sensing. IEEE Communications Magazine 2010;48(9):140–50.

La Polla M, Martinelli F, Sgandurra D. A survey on security for mobile devices. IEEE Communications Surveys & Tutorials 2012;15(1):446–71.

Leistikow R, Tavangarian D, 2013. Secure picture data partitioning for cloud computing services. In: 2013 27th International conference on advanced information networking and Applications Workshops, IEEE, 2013. p. 668–71. http://dx.doi.org/10.1109/WAINA.2013.157.

Loutas N, Peristeras V, Bouras T, Kamateri E, Zeginis D, Tarabanis K. Towards a reference architecture for semantically interoperable clouds. In: Proceedings of IEEE CloudCom'10, 2010. p. 143–50.

Lu Y, Li SP, Shen HF. Virtualized screen: a third element for cloud-mobile convergence. IEEE Multimedia 2011;18(2):4–11.

Lu W, Lu D, Lu M, Hu J. Open wireless architecture (OWA) unified airborne and terrestrial communications architecture, 2012.

Madhavapeddy A, Mortier R, Crowcroft J, Hand S. Multiscale not multicore: efficient heterogeneous cloud computing. In: Proceedings of ACM-BCS'10, 2010.

Mao H, Xiao N, Shi W, Lu Y. Wukong: a cloud-oriented file service for mobile Internet devices. Journal of Parallel and Distributed Computing 2011;72(2):171–84.

March V, Gu Y, Leonardi E, Goh G, Kirchberg M, Lee BS. μCloud: towards a new paradigm of rich mobile applications. In: Proceedings of IEEE MobWIS'11, Ontario, Canada, 2011.

Marinelli EE. Hyrax: cloud computing on mobile devices using MapReduce [Master thesis]. Computer Science Department, Carnegie Mellon University, 2009.

Marquardt P, Verma A, Carter H, Traynor P. (sp)iPhone: decoding vibrations from nearby keyboards using mobile phone accelerometers. In: Proceedings of ACM CCS'11, Chicago, IL, USA, 2011. p. 551–62.

Martín-Campillo A, Crowcroft J, Yoneki E, Martí R. Evaluating opportunistic networks in disaster scenarios. Journal of Network and Computer Applications 2013;36(2):870–80.

Meier JD, Hill D, Homer A, Jason Taylor, Bansode P, Wall L, et al. Microsoft application architecture guide. Microsoft Press Book; 2009.

Meliá S, Gómez J, Pérez S, Díaz O. Architectural and technological variability in Rich Internet Applications. IEEE Internet Computing 2010;14(3):24–32.

Miettinen AP, Nurminen JK. Energy efficiency of mobile clients in cloud computing. In: Proceedings of USENIX HotCloud'10, Boston, MA, 2010. p. 4–11.

Mukhopadhyay S, Schurgers C, Dey S. Enabling rich mobile applications: joint computation and communication scheduling. Mobile Computing and Communications Review 2009;13(3):14–25.

Murugesan S, Deshpande Y, 2002. Meeting the challenges of web application development: the web engineering approach. In: Proceedings of ACM ICSE'02, Orlando, FL, USA, 2002. p. 687–8.

Neuvo Y. Cellular phones as embedded systems. In: Digest of technical papers, IEEE ISSCC'04, vol. 47, 2004. p. 32–7.

Noda T, Helwig S. Rich internet application, Technical comparison and case studies of Ajax, flash and java based RIA. Technical Report, University of Wisconsin, Madison; 2005.

Norman D, Draper S. User centered system design: new perspectives on human–computer interaction. L. Erlbaum Associates Inc.; 1986.

Oehlman D, Blanc S. Pro android web apps, develop for android using HTML5, CSS3,& Java script, Apress, 2011.

Omar M, Challal Y, Bouabdallah A. Certification-based trust models in mobile ad hoc networks: a survey and taxonomy. Journal of Network and Computer Applications 2012;35(1):268–86.

Open Kernel Labs and Citrix Systems. The "nirvana phone": concept specification and draft reference architecture. Technical Report; 2009.

Ortiz S. Bringing 3D to the small screen. IEEE Computer 2011;44(10):11–3.

Panta RK, Jana R, Cheng F, Chen Y-FR, Vaishampayan Va. Phoenix: storage using an autonomous mobile infrastructure. IEEE Transactions on Parallel and Distributed Systems 2013;24(9):1863–73, http://dx.doi.org/10.1109/TPDS.2013.84.

Perrucci GP, Fitzek FHP, Widmer J. Survey on energy consumption entities on the smartphone platform. In: Proceedings of IEEE VTC Spring'11, Budapest, Hungary, 2011. p. 1–6.

Pickard WF, Abbott D. Addressing the intermittency challenge: massive energy storage in a sustainable future [Scanning the Issue]. Proceedings of the IEEE 2012;100(2):317–21.

Proebster M, Kaschub M, Valentin S. Context-aware resource allocation to improve the quality of service of heterogeneous traffic. In: Proceedings of IEEE ICC'11, 2011. p. 1–6.

Prosper Mobile Insights, Smartphone/tablet user survey; 2011. URL: ⟨http://prospermobileinsights.com/Default.aspx?pg=19⟩.

Rahimi MR. Exploiting an elastic 2-tiered cloud architecture for rich mobile applications. In: Proceedings of IEEE WoWMoM'12, IEEE, 2012. p. 1–2. http://dx.doi.org/10.1109/WoWMoM.2012.6263740.

Rahimi M, Venkatasubramanian N, Mehrotra S, Vasilakos V. MAPCloud: mobile applications on an elastic and scalable 2-tier cloud architecture. In: Proceedings of IEEE/ACM UCC'12, Chicago, Illionis, USA, 2012.

Ranabahu A, Sheth A. Semantics centric solutions for application and data portability in cloud computing. In: Proceedings of IEEE CloudCom'10, Indianapolis, USA, 2010. p. 234–41.

Sabou M. Smart objects: challenges for semantic web research. Semantic Web 2010;1(1):127–30.

Sanaei S, Abolfazli Z, Shiraz M, Gani A. SAMI: service-based arbitrated multi-tier infrastructure model for mobile cloud computing. In: Proceedings of IEEE MobiCC'12, Beijing, China, 2012. p. 14–9.

Sanaei Z, Abolfazli S, Gani A, Khokhar RH. Tripod of requirements in horizontal heterogeneous mobile cloud computing. In: Proceedings of WSEAS CISCO'12, Singapore, 2012.

Sanaei Z, Abolfazli S, Gani A, Buyya RK. Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Communications Surveys Tutorials 2013.

Satyanarayanan M. Avoiding dead batteries. IEEE Pervasive Computing 2005;4(1):2–3.

Satyanarayanan M, Bahl P, Caceres R, Davies N. The Case for VM-based cloudlets in mobile computing. IEEE Pervasive Computing 2009;8(4):14–23.
Sharifi M, Kafaie S, Kashefi O. A survey and taxonomy of cyber foraging of mobile devices. IEEE Communications Surveys & Tutorials 2011;99:1–12.
Shiraz M, Abolfazli S, Sanaei Z, Gani A. A study on virtual machine deployment for application outsourcing in mobile cloud computing. The Journal of Supercomputing 2012c;63(3):946–64.
Song JQ, Tang EY, Liu LB. User behavior pattern analysis and prediction based on mobile phone sensors. Network and Parallel Computing 2010;6289:177–89.
Srivastava S, Nandi G. Self-reliant mobile code: a new direction of agent security. Journal of Network and Computer Applications 2013.
Starner T, Kirsch D, Assefa S. The locust swarm: an environmentally-powered, networkless location and messaging system. In: Proceedings of IEEE ISWC' 97, Boston, MA, USA, 1997. p. 169–70.
Stuedi P, Mohomed I, Terry D. WhereStore: location-based data storage for mobile devices interacting with the cloud. In: Proceedings of ACM MCS'10, San Francisco, California, USA, 2010. p. 1.
Thomas RW, DaSilva LA, MacKenzie AB. Cognitive networks. In: Proceedings of IEEE DySPAN'05, Baltimore, MD, USA, 2005. p. 352–60. http://dx.doi.org/10.1109/DYSPAN.2005.1542652.
Tolia N, Andersen DG, Satyanarayanan M. Quantifying interactive user experience on thin clients. IEEE Computer 2006;39(3):46–52.
Truong HL, Dustdar S. A survey on context-aware web service systems. International Journal of Web Information Systems 2009;5(1):5–31.
Valberg S, Christensen PT. Improving usability of mobile devices by means of accelerometers [Ph.D. thesis]. Technical University of Denmark, 2009.
Vallina-Rodriguez N, Crowcroft J. Energy management techniques in modern mobile handsets. IEEE Communications Surveys & Tutorials 2013;15(1):179–98.
Verbelen T, Simoens P, De Turck F, Dhoedt B. AIOLOS: middleware for improving mobile application performance through cyber foraging. Journal of Systems and Software 2012;85(11):2629–39.
von Mulert J, Welch I, Seah WK. Security threats and solutions in MANETs: a case study using AODV and SAODV. Journal of Network and Computer Applications 2012;35(4):1249–59.
Wacharasindhu T, Kwon JW, Meier DE, Robertson JD. Liquid-semiconductor-based micro power source using radioisotope energy conversion. In: IEEE international conference on solid-state sensors, actuators and microsystems, Denver, CO, USA, 2009. p. 656–9.
Wang C, Wang Q, Ren K, Lou W. Ensuring data storage security in cloud computing. In: Proceedings of IEEE IWQoS'09, Charleston, South Carolina, USA, 2009. p. 1–9.
Wang T, Cardone G, Corradi A, Torresani L, Campbell AT. WalkSafe: a pedestrian safety app for mobile phone users who walk and talk while crossing roads. In: Proceedings of ACM HotMobile'12, San Diego, California, USA, 2012.
Webinar: designing RMAs in a fragmented world; 2011. URL: ⟨http://worklight.com/resources/webinar-designing-rich-mobile-apps⟩.
West MT. Ubiquitous computing. In: Proceedings of ACM SIGUCCS'11, 2011. p. 175–82.
Whaiduzzaman M, Sookhak M, Gani A, Buyya R. A survey on vehicular cloud computing. Journal of Network and Computer Applications 2013, http://dx.doi.org/10.1016/j.jnca.2013.08.004.
Whitten A. Making security usable. Ph.D. thesis. Princeton University, 2004.
Woerndl W, Schueller C, Wojtech R. A hybrid recommender system for context-aware recommendations of mobile applications. In: Proceedings of ICDE'07, IEEE, Istanbul, Turkey. p. 871–8.
Wu H-K, Lee SW-Y, Chang H-Y, Liang J-C. Current status, opportunities and challenges of augmented reality in education. Computers & Education 2013;62:41–9.
Yih-Farn C, Gibbon D, Jana R, Renger B, Stern D, Yang M, et al. Project GeoTV—a three-screen service: navigate on SmartPhone, browse on PC, watch on HDTV. In: Proceedings of IEEE CCNC'09, Las Vegas, NV, USA, 2009. p. 1–2.
Yin W, Luo J, Chen CW. Event-based semantic image adaptation for user-centric mobile display devices. IEEE Transactions on Multimedia 2011;13(3):432–42.
Yu P, Ma X, Cao J, Lu J. Application mobility in pervasive computing: a survey. Pervasive and Mobile Computing 2013;9(1):2–17.
Yuksel KA, Kehribar I, Ercil A. Touchless magnetic interaction with mobile devices. In: Proceedings of IEEE SIU'11, Antalya, Turkey, 2011. p. 936–9.
Zhang XW, Kunjithapatham A, Jeong S, Gibbs S. Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing. Mobile Networks & Applications 2011a;16(3):270–84.
Zhang D, Huang H, Lai CF, Liang X, Zou Q, Guo M. Survey on context-awareness in ubiquitous media. Multimedia Tools and Applications 2011b:1–33.
Zheng W, Xu P, Huang X, Wu N. Design a cloud storage platform for pervasive computing environments. Cluster Computing 2010;13(2):141–51.